**Performance Analysis of AODV Protocol in MANETs: Impact of Node Density and Transmission Range on Packet Delivery Ratio and Throughput using NS-3**

Completed by:

Anurag Patel (121CS0201)

Rajdeep Das (121CS0160)

Md Nafis Al Safayet (121CS0217)

Tanish Raj Purty (121CS0206)

Anubhuti Pandey (121CS0200)

**Network Simulation Laboratory**

**Submitted on:**



**CONTENT**

Page No.

1. Introduction 3
   1. Background 3

2. Flowchart and Algorithm 4

2.1. Flowchart 4

2.2. Algorithm 6

3. Results and Discussion 7

3.1. Simulation Setup 7

3.2. Performance Analysis 8

3.2.1. Throughput 8

3.2.2. End to End Delay 9

3.3.3. Packet Loss 10

4. Conclusion 11

5. Manual 12

6. References 12

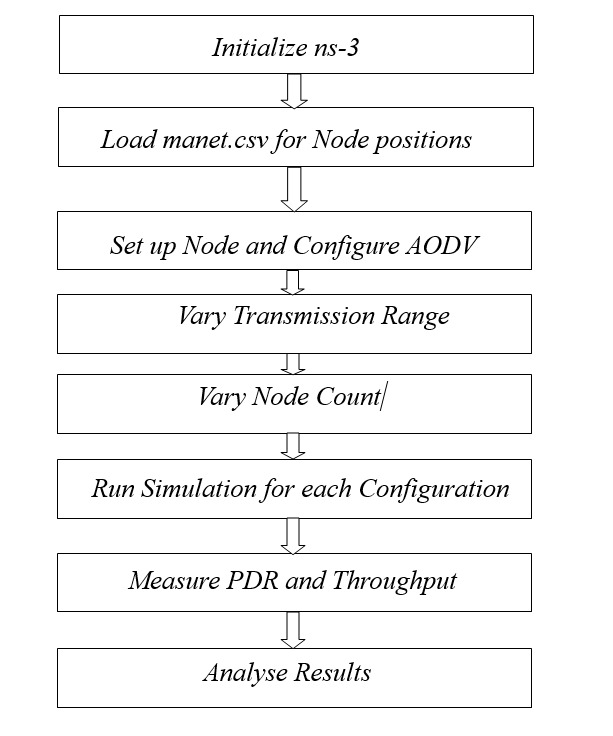
1. **Introduction**
   1. **Background**

The Ad hoc On-Demand Distance Vector (AODV) protocol is widely used in Mobile Ad hoc Networks (MANETs) due to its efficient route discovery and maintenance. Unlike traditional protocols, AODV dynamically establishes routes only when requested by source nodes, minimizing network overhead. In this project, using the ns-3 simulator, we explore AODV’s performance across a network of 100 nodes with positions set by the manet.csv file. By adjusting transmission ranges and node densities, we aim to measure packet delivery ratio and throughput, analysing the network's response to changing parameters.

**2. Flowchart and Algorithm**

**2.1. Flowchart**

The flowchart can be given as follows:



**Initialize ns-3:** Set up the environment for the ns-3 simulator. It includes simulating parameters such as simulation time and initial set-up for the usage of the AODV protocol.

1. **Load “manet.csv” for Node Positions:** Load the file \*manet.csv that has the X-Y coordinates of the 100 nodes for this simulation. In this simulation, these locations will dictate the spatial organization of nodes in a network.
2. **Set up Node and Configure AODV:** Now, loading node positions, simulating the system with each node involved, and using AODV as a routing protocol ensures that AODV takes control of the route for packet transmission over the data and the discovery over the nodes.
3. **Vary Transmission Range:** In this case, changes regarding the range through which a particular node can transmit its data occur. The kinds of change will determine the type of change in node connectivity. It is also the only way through which change of node connectivity determines a new change in the network.
4. **Vary Node Count:** Vary the node density of the network for distinct simulation runs. Node density needs to be increased or reduced in order to check network load effects on the AODV performance.
5. **Run Simulations for each Configuration:** For each and every combination of transmission range and number of nodes run the simulation. It must require several simulation runs with distinguished configurations in order to capture various network conditions.
6. **Measure PDR and Throughput**: Use the monitor, ns3::FlowMonitor, within ns-3 to write out Packet Delivery Ratio and throughput after each simulation run. These measures will allow one to contrast how AODV is performing in this simulated setting.
7. **Analyse Results:** After all simulations are performed, analyse the data obtained. Observe the trend of packet delivery ratio and throughput for different transmission ranges and node densities. This analysis will give an insight into how AODV performs in changing network conditions.

**2.2. Algorithm**

**a. Route Discovery (On-Demand)**

* Initiates route discovery only when a source node requires a path to a destination.
* Sends a Route Request (RREQ) packet to neighbouring nodes.

**b. Broadcasting RREQ**

* Each node receiving the RREQ forwards it to its neighbors if it doesn’t have a valid route to the destination.
* RREQ propagation continues until it reaches the destination or a node with a fresh route to the destination.

**c. Route Reply (RREP)**

* The destination node, or an intermediate node with a fresh route, sends a Route Reply (RREP) back to the source.
* RREP traverses back along the path established by RREQs, updating routing tables en route.

**d. Route Maintenance**

* Routes remain active as long as they are being used.
* If a link break occurs, the affected node sends a Route Error (RERR) to inform upstream nodes.

**e. Sequence Numbers for Freshness**

* AODV uses sequence numbers to ensure routes are loop-free and up-to-date, allowing nodes to select the most recent route.

**f. Periodic Hello Messages**

* Nodes broadcast hello messages periodically to maintain connectivity with neighbours.

This algorithm ensures efficient, on-demand routing in dynamic, decentralized networks, minimizing routing overhead and adapting quickly to network topology changes.

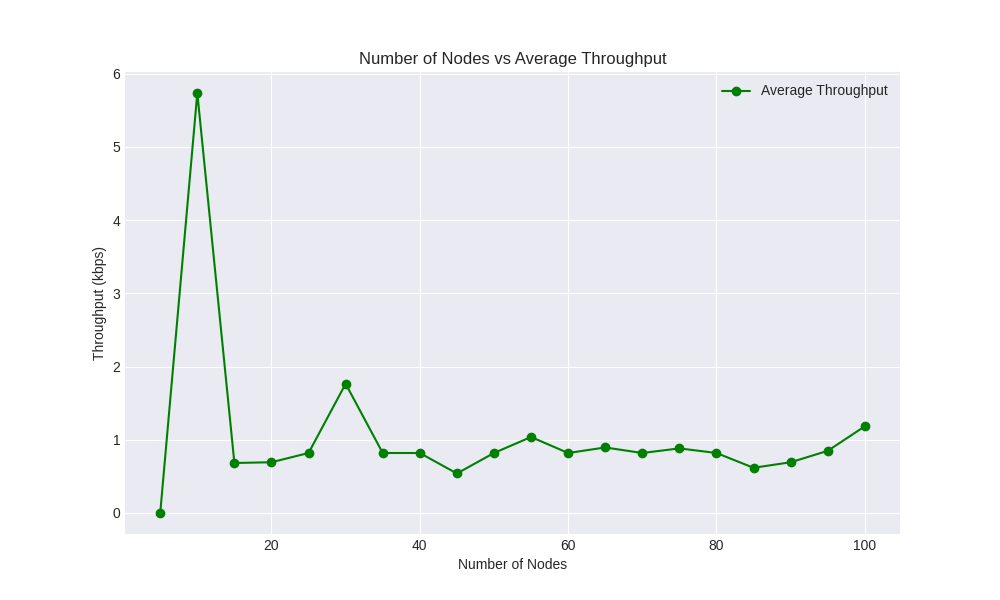
**3. Results and Discussion**

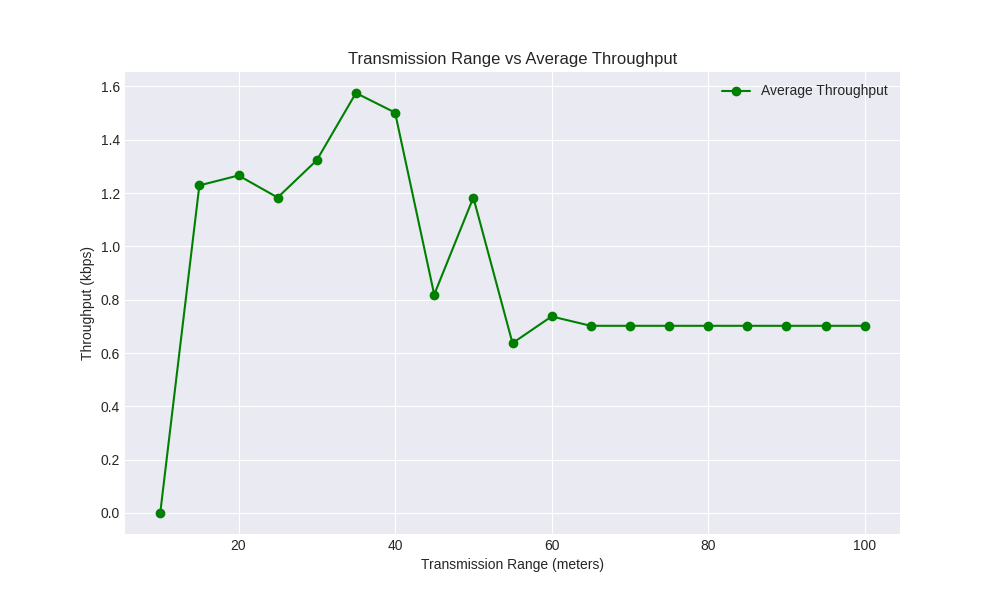
**3.1. Simulation Setup**

* **Version of ns3**: ns-3.35
* **Total Source Nodes**: 1 (assuming a total of 100 nodes)
* **Total Destination Nodes**: 99 (assuming a total of 100 nodes)
* **Number of Routers**: 100
* **Queue Management**: DropTailQueue.
* **Link Management**: **YansWifiChannel** and **RangePropagationLossModel.**
* **Bandwidth: 500 Mbps.**
* **Delay: No Explicit Delay is given.**

**3.2. Performance Analysis**

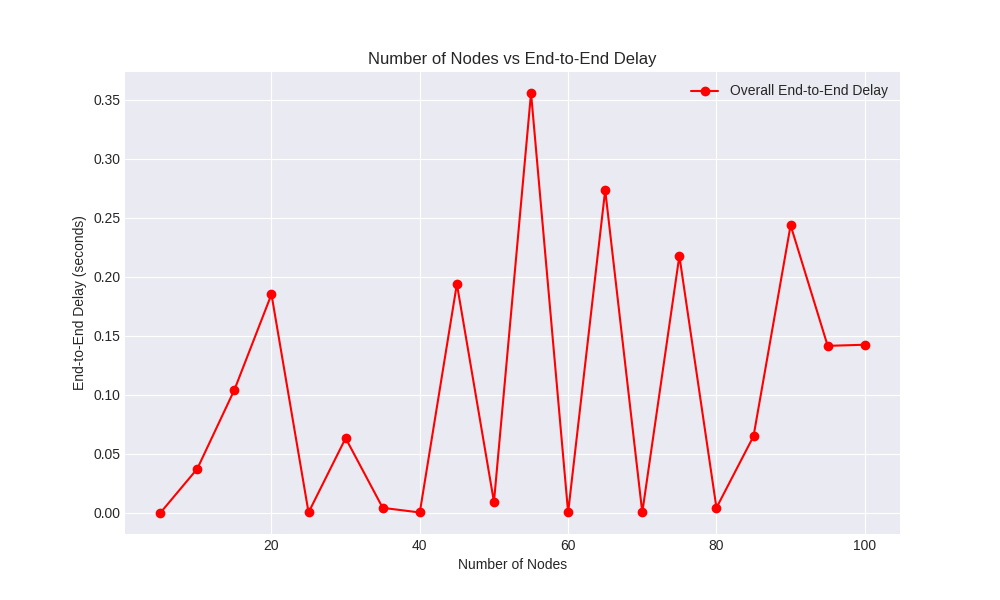
**3.2.1. Average Throughput**

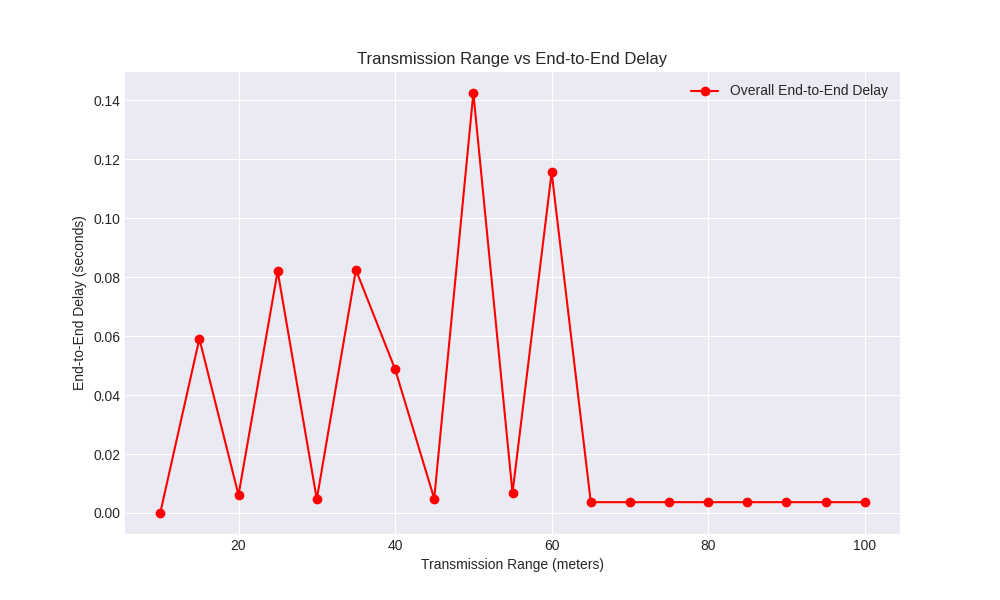
****

****

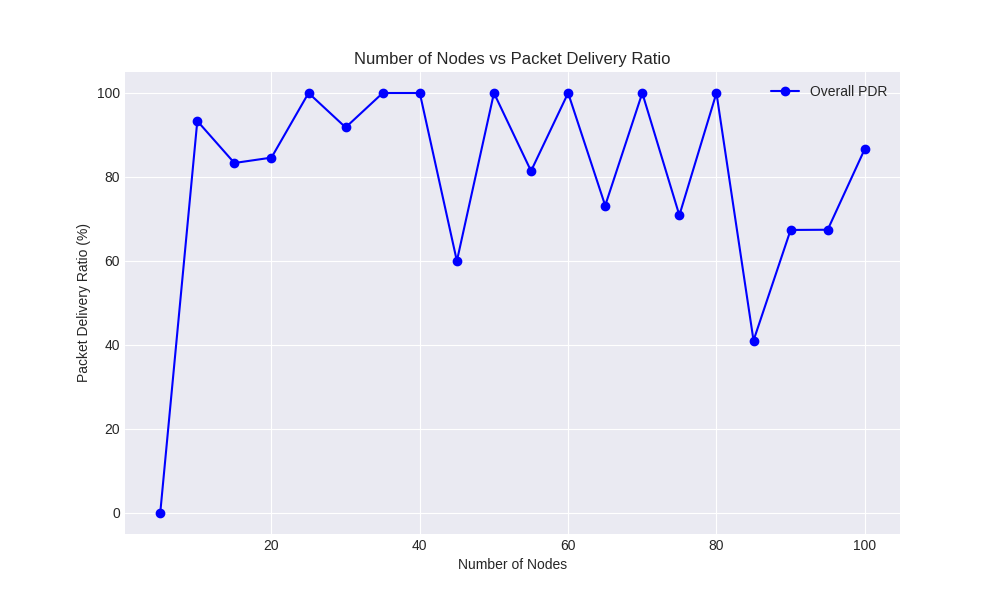
The graphs illustrate the relationship between average throughput (kbps) and two key network parameters: the number of nodes and transmission range. In the "Number of Nodes vs Average Throughput" graph, throughput initially spikes with fewer nodes, likely due to minimal contention, but stabilizes as node count increases, indicating network congestion. In the "Transmission Range vs Average Throughput" graph, throughput rises up to an optimal range, possibly due to better connectivity, but decreases as range continues to increase, potentially due to interference or signal attenuation.

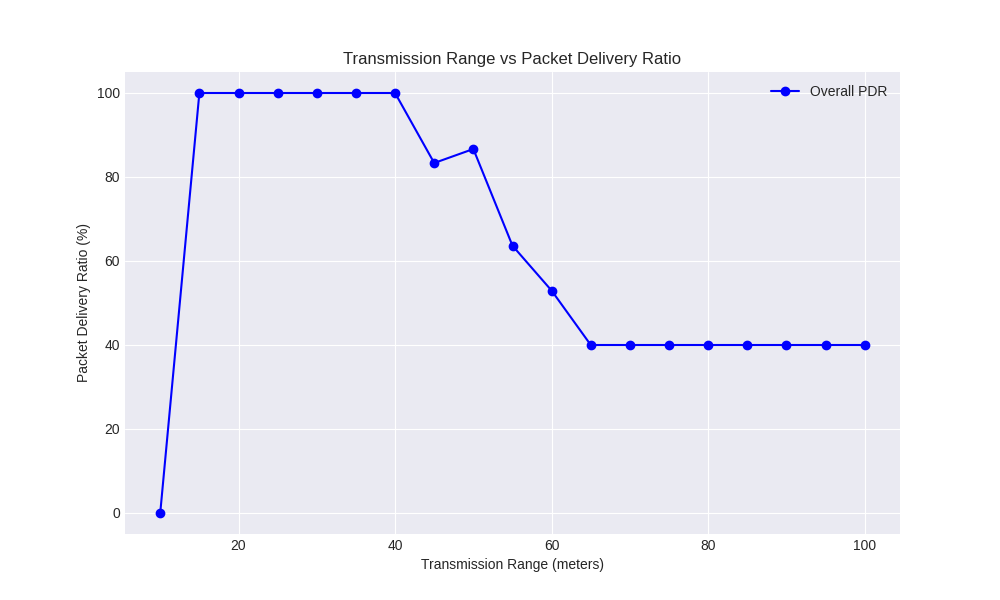
**3.2.2 End to End Delay**

****

****

The first graph shows fluctuating end-to-end delay with increasing node numbers, while the second graph indicates decreasing delay as transmission range increases, improving connectivity.

**3.2.3 Packet delivery Ratio**

****

The first graph shows the relationship between the number of nodes and the packet delivery ratio (PDR), indicating stable performance up to 100 nodes with occasional dips. The second graph illustrates how the transmission range affects PDR, with optimal performance up to around 40 meters before significant drops.

**4. Conclusion**

We can arrive at the following conclusions after the completion of the project:

* + **Optimal Transmission Range**: Transmission ranges of **15-40 meters** yielded high packet delivery ratios (PDR) with low delay. Beyond 45 meters, PDR and throughput declined due to increased interference.
  + **Node Density Impact**: The network performed best with **10-40 nodes,** maintaining high PDR and stable throughput. **Densities above 50 nodes** led to reduced PDR and higher delays, showing AODV’s limitations in highly dense networks.
  + **AODV’s Suitability**: AODV is effective for **small to medium-sized networks** with moderate node density but struggles with scalability and performance in larger, denser environments.
  + **Protocol Limitations**: AODV’s on-demand routing mechanism is prone to increased overhead and latency in large or dense networks, impacting its scalability.

**5. Manual**

#### **Prerequisites:**

1. **NS-3 Installation**: Make sure NS-3 (version 3.35 or compatible) is installed on your system. You can download it from the NS-3 website.
2. **Environment Setup**:
   * Your experiment files (e.g., aodv\_m\_test.cc) and manet.csv file with node positions should be located in the scratch directory within your NS-3 installation.
   * Ensure manet.csv is correctly formatted with columns for node numbers, x-coordinates, and y-coordinates.

#### **Experiment Files:**

* **Simulation File**: aodv\_m\_test.cc - the primary source code for the AODV simulation.
* **Node Position File**: manet.csv - CSV file defining X-Y positions for each node.

### ***Steps to Run the Experiment:***

1. **Navigate to NS-3 Directory**: Open a terminal and navigate to the root directory of your NS-3 installation:
2. **Place Files in the Scratch Directory**: Ensure aodv\_m\_test.cc and manet.csv are in the scratch directory:
3. **Build the Simulation File**: Run the following command to compile the code:

./waf build --run scratch/aodv\_m\_test

* + If the build completes without errors, it will automatically run the simulation.
  + If there are errors, check for missing libraries or syntax issues in the code.

1. **Configure Command-Line Parameters**: Optional parameters can be set directly in the command line:
   * **Example**: Adjusting node count, transmission range, and simulation time.

./waf --run "scratch/aodv\_m\_test --size=100 --transmissionRange=50 --time=10"

* + Parameters available:
    - --size: Number of nodes (default is 100).
    - --transmissionRange: Transmission range in meters (default is 50).
    - --time: Total simulation time in seconds (default is 10).

1. **Output Files**: The simulation generates several output files:
   * **PCAP Traces** (if enabled): Stored as aodv-xx.pcap files.
   * **NetAnim File**: aodv\_simulation.xml - use this file for network animation with NetAnim.
   * **Flow Monitor XML**: flow-monitor-results.xml - for detailed flow statistics.
   * **CSV Results**: results.csv - containing Packet Delivery Ratio (PDR), average throughput, and end-to-end delay.
2. **Analyze Results**:
   * After the simulation completes, open the *results.csv* file to view the analyzed results.
3. **Run Simulation:**

Running simulation by varying number of nodes and transmission range:

* Keep the simul.py in the ns 3.35 directory and run the python file by command:

pyhton3 simul.py

* It will give a file results.csv which will contain the dataset for further analysis for varying nodes. Rename the file to *results\_by\_varying\_nodes.csv* and save it.
* Similarly keep the simul1.py in the ns 3.35 directory and run the python file by command

python3 simul1.py

* It will give a file *transmission\_range\_results.csv* which will contain the dataset for further analysis for varying nodes

1. To obtain the graph:

* To get the graphs of No. of nodes vs. PDR, No. of nodes vs. Average throughput, and No. of nodes vs. end to end delay run ns3\_results\_1.py file.
* To get the graphs of Transmission\_range vs. PDR, Transmission\_range vs. Average throughput, and Transmission\_range vs. end to end delay run ns3\_results\_2.py file.

**6. References:**

* <https://www.researchgate.net/publication/325512294_STATIC_NETWORK_PERFORMANCE_OPTIMISATION_USING_DUAL_INTERLEAVE_ROUTING_ALGORITHM>
* <https://www.researchgate.net/publication/330180796_Comparison_of_AODV_DSR_and_DSDV_routing_protocols_in_a_wireless_network>
* <https://www.nsnam.org/documentation/>