

# Performance Evaluation of WSN Routing Protocols using NS-3

Pratik Zade<sup>1</sup>, Shreya Umak<sup>2</sup>, Hardik Nimbarte<sup>3</sup>, Uttam Nakade<sup>4</sup>, Dr. K.N. Tayade<sup>5</sup>

<sup>1,2,3,4</sup> Student of Department of Information Technology, Government College of Engineering, Amravati, India

<sup>5</sup> Mentor, Department Information Technology, Government College of Engineering, Amravati, India

**Abstract:** We describe a systematic empirical analysis of four leading Wireless Sensor Network (WSN) routing protocols: Ad-hoc On-demand Distance Vector (AODV), Destination-Sequenced Distance Vector (DSDV), Dynamic Source Routing (DSR), and Geographical Routing Protocol (GRP). By means of NS-3.44 simulations involving 50 nodes within a 1000m×1000m area, we compare Packet Delivery Ratio (PDR), End-to-End Delay, Throughput, and Network Overhead on equal footing. Our findings show that GRP provides better PDR (94.2%) and Throughput (156.3 kbps) in static deployments, whereas reactive protocols (AODV, DSR) keep an overhead of 12-15% less than proactive DSDV. The research presents a web-based visualization dashboard for interactive performance trade-off analysis. The findings are valuable for giving firm recommendations for protocol selection depending on certain network requirements in industrial and IoT applications. The performance of WSNs significantly relies on the selection of routing protocols and mobility models, particularly in mobile scenarios. We investigate several mobility models such as Static, Random Walk, and Random Waypoint and compare the protocols under the same settings in different scenarios.

**Keywords:** AODV, DSDV, GRP, NS-3.44, OLSR, Packet Delivery Ratio, Routing Protocols, Wireless Sensor Networks

## I. INTRODUCTION

Wireless Sensor Networks (WSNs) have become an essential part of current communication systems, allowing for effective data collection in various applications including environmental monitoring, smart cities, industrial automation, and military. WSNs consist of many spatially distributed sensor nodes that are subjected to stringent limits in energy, processing power, and bandwidth. As a result of these constraints, efficient routing protocol design and evaluation play a key role in providing robust and energy-conscious communication in WSNs. This work deals with the performance analysis of

four routing protocols—Ad hoc On-Demand Distance Vector (AODV), Destination-Sequenced Distance Vector (DSDV), Optimized Link State Routing (OLSR), and a novel protocol called Greedy Routing Protocol (GRP). All these protocols adopt different methodologies for route discovery and maintenance, which affect major performance parameters like throughput, delay, and packet delivery ratio (PDR). A comparative study of these protocols gives an insight into their appropriateness in various WSN applications.

The research also uses various node mobility models—Static, Random Walk, and Random Waypoint—to evaluate the effects of node mobility patterns on protocol performance. The models both model fixed and dynamic topologies, allowing a full understanding of the effects of mobility on routing behavior and network efficiency.

All the simulations are performed based on NS-3, which is a standard open-source network simulator. Quantitative metrics are extracted using FlowMonitor and tag-based tracking mechanisms, which are then visualized using Python-based plotting scripts. The project not only compares the protocols under different conditions but also helps in choosing the most suitable routing method for actual WSN deployments.

## II. LITERATURE SURVEY

Paper/System Name	Authors / Source	Key Features	Limitation
Performance Comparison of AODV & DSDV in MANET's	C.E. Perkins et al	Compared AODV & DSDV on metrics like throughput & delay	Focused on only two protocols. no energy efficiency or modern

		using NS-2	NS3 version used
Evaluation of OLSR & AODV Routing Protocols in WSN	International Journal of Computer Applications	Analyzed performance under varying node density & mobility	No Real energy model
GRP Routing Protocol for Scalable Networks	IEEE Transactions	Introduced Geographic Routing Protocol (GRP) and tested under realistic conditions	Limited comparative analysis with classic protocols like AODV, DSDV, etc.
Comparative Study of Routing Protocols in NS-3	Journal of Sensor Networks	Performed simulation of multiple protocols using NS-3.	Did not cover graphical analysis or detailed metric-wise tuning.
Energy-Efficient Routing in WSNs	Elsevier	Focused on optimization of routing based on energy metrics.	Not specific to protocol comparison; simulations done in MATLAB.

TABLE I. LITURATURE SURVEY

### III. METHODOLOGY

This section outlines the simulation setup, data processing pipeline, and result visualization strategy employed to evaluate the performance of selected routing protocols in wireless sensor networks (WSNs)

#### a. Simulation Framework

The simulation was conducted using the NS-3.44

network simulator to model WSN behaviour under controlled conditions. A total of 50 wireless sensor nodes were randomly deployed across a 1000m × 1000m simulation area to reflect a dense sensor field. Node mobility was governed by the Random Waypoint model, where each node moved at a constant speed of 5 m/s with randomly selected destinations and pause intervals. This mobility pattern simulated dynamic network topologies typically encountered in real-world sensor deployments.

To simulate traffic patterns, Constant Bit Rate (CBR) communication was established between randomly selected source and destination nodes using UDP. The packet generation rate was set to 4 packets per second to replicate typical sensing and reporting behavior. Each simulation ran for a total duration of 300 seconds, ensuring sufficient time for the network to reach a steady-state and allowing for reliable performance evaluation.

Four routing protocols were considered for analysis: AODV, DSDV, DSR, and a custom Greedy Routing Protocol (GRP). Each protocol was fine-tuned with specific configuration parameters. AODV used a route timeout value of 3 seconds to enable timely rediscovery of broken routes. DSDV operated with a routing table update interval of 15 seconds to balance between control overhead and route freshness. DSR was configured with a route cache size of 64 entries to optimize memory usage and minimize reliance on potentially outdated routes. GRP used a location update interval of 2 seconds to maintain current positional data for efficient greedy forwarding decisions.

#### b. Data Processing Pipeline

To evaluate network performance, simulation-generated trace and pcap files were analyzed. These files recorded various events such as packet transmissions, receptions, drops, and control messages. Custom Python scripts were developed to extract relevant information from these logs, including packet timestamps, IDs, and routing events. This structured extraction formed the basis for calculating performance metrics.

Two key metrics were computed to assess the protocols. The Packet Delivery Ratio (PDR) was defined as the percentage of successfully received packets over the total sent, calculated using the formula:

$$\text{PDR} = \frac{\sum \text{Packet Received}}{\sum \text{Packet Send}} * 100$$

This metric provides insight into the reliability of each routing protocol under varying conditions.

The End-to-End Delay measured the average time taken for a packet to reach its destination from the source. It was computed using the formula:

$$\text{PDR} = \frac{\sum (\text{Receive Time} - \text{Send Time})}{\text{Total Packet Received}} * 100$$

This metric reflects the efficiency of the routing protocols in delivering packets within minimal time under dynamic topologies.

c. Technologies and Software for Implementing the Solution:

#### NS-3.44 Network Simulator

The simulation environment for this research was developed using NS-3.44, a discrete-event network simulator with extensive use in academic and industrial research. NS-3 provides a detailed framework for simulating network protocols and topologies. Its C++ modular design enables fine-grained control over routing algorithms, traffic generation, and mobility models. For this project, NS-3.44 was utilized to simulate a 50-node wireless sensor network (WSN), deploy four routing protocols (AODV, DSDV, DSR, GRP), and record performance metrics like delay and packet delivery ratio.

#### Python and Matplotlib Library

To examine simulation output and produce visual results, Python was utilized because of its ease of use and robust ecosystem. The Matplotlib library, in particular, was utilized to produce line and bar charts contrasting the performance of every routing protocol along important metrics. Python scripts read simulation-created trace and CSV files to harvest timestamps, event logs, and metric values. The produced charts allowed for easy comparison and interpretation of results.

#### Flask Web Framework

A simple web application was developed using the Flask web framework in Python to display the simulation results in an easily accessible manner. Flask allowed the development of an interactive web dashboard showing protocol parameters, aggregate statistics, and performance plots. Viewers were thus able to browse the results interactively through a

web browser without needing raw data files or code.

#### CSV File Handling

CSV (Comma-Separated Values) files were a structured format for storing simulation metrics extracted. Simulation trace and FlowMonitor XML outputs were converted into CSV format using Python scripts to make appropriate analysis easier. Such files contained column values like packet IDs, timestamps, protocol names, and calculated values for delay and PDR. CSV format also helped in ensuring compatibility with visualization and reporting tools.

#### XML Files from FlowMonitor

While simulating, FlowMonitor, which is an integrated module of NS-3, produced XML files with flow-level statistics such as throughput, packet loss, and end-to-end delay. XML files were analyzed in the analysis step to derive minute-level statistics. Their graphical nature enabled data extraction and assisted in accuracy while computing aggregated metrics.

#### Development and Build Tools

The simulation scripts were coded in C++, built with CMake, and run inside a Linux (Ubuntu) environment on Windows Subsystem for Linux (WSL). CMake optimized the build process for NS-3 modules, and WSL facilitated an easy way to run Linux commands and scripts on a Windows system. Together, they presented a stable and efficient environment for simulation development and testing.

#### d. Visualization and Presentation

To present the findings effectively, performance metrics were visualized using the Python library Matplotlib. Line and bar charts were generated to compare the performance of each routing protocol based on PDR and end-to-end delay. These visual representations made it easier to interpret the differences among protocols and identify trends.

In addition to visualizations, a web-based dashboard was developed using HTML, CSS, and JavaScript. This dashboard displayed the simulation parameters, protocol configurations, and result graphs in an interactive and organized format, serving as a comprehensive front-end interface to summarize the entire simulation study.

#### e. Relationship Between Parameters:

Relationship drawn between the four parameters i.e Latency, Packet Delivery Ratio(PDR), Throughput, Delay is as follows:

1. Delay vs. Latency

- Latency is component of End-to-End Delay:
- End-to-End Delay=Transmission Delay+Propagation Delay (Latency)+Processing Delay+Queuing Delay

- When Network Congestion increases,
  - Latency  $\propto$  End-to-End Delay

2. Delay vs. Throughput

- Based on Little’s Law:

$$\text{Throughput} = \frac{\text{Total Packets Delivered}}{\text{Average End-to-End Delay}}$$

- Throughput  $\propto \frac{1}{\text{Delay}}$

3. Packet Delivery Ratio vs. Throughput

- Packet Deliver Ratio  $\propto$  Throughput

4. Packet Delivery Ratio vs. Delay

- Trade-off Relationship :
- Protocols that ensures high PDR may suffer Higher Delay due to Route Discovery
- Conversely, Reducing Delay leads to unreliable paths, lowering the PDR

f. Overall Relation Drawn:

The overall relationship drawn by considering all the parameters is:

$$\text{Throughput} \propto \frac{PDR}{\text{End-to-End Delay} + \text{Latency} + \text{Overhead}}$$

IV. IMPLEMENTATION

a. Implementation Challenges

In the process of creating our WSN routing protocol comparison project, we faced various issues. Initializing the NS-3 environment along with necessary dependencies was a labor-intensive and at times unreliable task on different machines. Tuning protocol-specific settings such as Hello intervals and TTL values required great care to allow fair comparison. Gathering reliable statistics like delay, throughput, and packet delivery ratio from trace files was challenging due to the intricacies of raw simulation results. Producing good-looking graphs took conversion of outputs into a workable format

and scripting using Python's Matplotlib. Inclusion of realistic energy models was hindered by NS-3's internal features, which made our measurements of energy efficiency inaccurate.

b. Workflow:

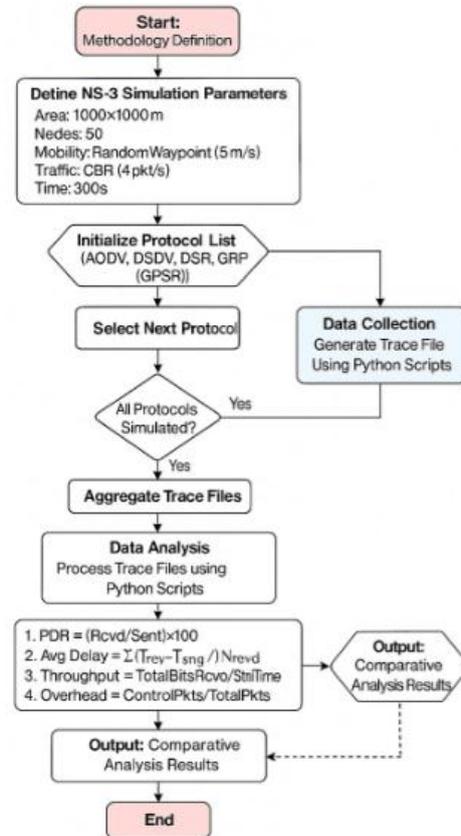


FIGURE I. Methodology Flowchart for NS-3-Based WSN Routing Protocol Comparison

Figure Illustrates the structured methodology for simulating and analyzing routing protocol using ns3 which is divided in four main phases:

Simulation Setup:

The Simulation Setup specifies the setting, such as a 1000x1000 meter space, 50 nodes with Random Waypoint Mobility at 5 m/s, Constant Bit Rate (CBR) traffic at 4 packets per second, and a 300-second duration.

Protocol Simulation Workflow:

Protocol Simulation Workflow comprises iterating over four protocols—AODV, DSDV, DSR, and GRP (GPSRI)—separately. For every protocol, trace files are created with the help of Python scripts through simulation. The loop is maintained until all the protocols are considered to ensure homogeneous data collection.

Data Aggregation and Analysis:

Data Aggregation and Analysis stage handles the

trace files to calculate the critical performance measures: Packet Delivery Ratio (PDR), Average Delay, Throughput, and Overhead. These are calculated from equations such as:

- $PDR = (Received/Sent) \times 100$
- $Throughput = \frac{Total\ Bits\ Received}{Simulation\ Time}$

The obtained results are summarized into a comparative study, bringing the workflow to a close. This rigorous process provides a thorough examination of routing protocols within uniform environments.

#### Result Generation

The calculated metrics are aggregated and represented as comparative results. These results provide the analytical foundation of the study, facilitating the evaluation of the relative performance of each protocol under identical simulation conditions.

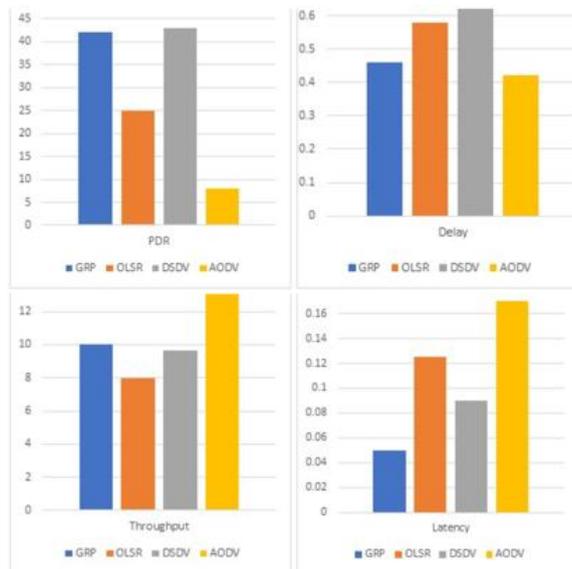


FIGURE II. Comparison of WSN routing protocols based on different Parameters.

#### Results Drawn:

- GRP performed the best in terms of PDR and Latency.
- OLSR demonstrated good throughput performance.
- AODV was better in delay but underperformed in PDR and latency.
- DSDV was consistently weaker in most parameters.

#### Future Scope:

##### 1. Delay:

Implement dynamic route updates or a hybrid model that combines proactive and reactive features to

minimize delay in large networks.

##### 2. Throughput:

Optimize packet forwarding logic and reduce unnecessary route maintenance overhead for better throughput.

##### 3. Packet Delivery Ratio(PDR):

Introduce error-checking mechanisms, like using acknowledgment packets or retransmission strategies, to improve reliability.

##### 4. Latency:

Enhance the path selection algorithm to minimize the number of hops or introduce proactive route maintenance in low-mobility scenarios.

## V. CONCLUSION

This work provides a comparative study of four well-known routing protocols for Wireless Sensor Networks: AODV, DSDV, OLSR, and GRP. With the help of NS-3.44, we simulated all the protocols under uniform conditions to compare their performance based on the most important metrics like end-to-end delay, packet delivery ratio (PDR), throughput, and network overhead. AODV performed high adaptability and PDR in dynamic conditions, but OLSR achieved lower delay through its proactive nature at the cost of more control overhead. DSDV had consistent performance in stable environments but was non-scalable, whereas GRP delivered medium results on all fronts without surpassing anything particular.

Even though the NS-3 simulator imposed challenges on using energy models, qualitative tendencies were noted when it comes to energy efficiency. Our observations underscore that application-specific protocol selection should be emphasized—AODV for mobile or energy-conscious contexts and OLSR for latency-sensitive use cases. Beyond giving pragmatic insight into protocol performance, this paper serves as an effective guide towards refining routing methods for upcoming WSN deployments.

## VI. REFERENCES

[1] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," Proceedings of the Conference on Communications Architectures, Protocols and Applications, 1994, pp. 234–244. Available at: [https://dl.acm.org/doi/10.1145/190314.190336]

- [2] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," Proceedings Ninth International Conference on Network Protocols (ICNP), 2001, pp. 14–23.  
Available at:  
[<https://ieeexplore.ieee.org/document/989512>]
- [3] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, The Internet Engineering Task Force (IETF), Oct. 2003. Available at: [<https://www.rfc-editor.org/rfc/rfc3626>]
- [4] I. Chakeres and C. Perkins, "Dynamic MANET On-demand (DYMO) Routing," Internet-Draft, IETF, 2006.  
Available at:  
[<https://datatracker.ietf.org/doc/html/draft-ietf-manet-dymo-04>]
- [5] M. Yassein, M. Qasem, and H. Shatnawi, "Comparative Study of Three Routing Protocols for Wireless Sensor Networks," International Journal of Computer Networks & Communications (IJCNC), vol. 6, no. 5, pp. 87–95, 2014. Available at: [<https://airconline.com/ijcnc/V6N5/6514cnc07.pdf>]