Volume: 13 Issue: IX Month of publication: September 2025
DOI: https://doi.org/10.22214/ijraset.2025.74032

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Interactive Visualization of Dynamic Source Routing Using a Flask-Based Web Framework

P Sai Prabhallika S[1], Prakash O. Sarangamath S [2], Dr. SDN Hayath Ali S[3]
Department of MCA, Ballari Institute of Technology and Management Ballari

Abstract: Dynamic Source Routing (DSR) has long served as a robust protocol for mobile ad hoc networks (MANETs), offering on-demand, adaptive routing ideal for infrastructure-less environments. However, traditional approaches to teaching and experimenting with DSR remain confined to backend simulations with limited visibility and interactivity. This addresses that gap by introducing an interactive, Flask-based web application designed to simulate and visualize the internal mechanics of the DSR protocol. The platform enables users to load custom network topologies, initiate route discovery, and observe real-time propagation of Route Request (RREQ), Route Reply (RREP), and error handling. The proposed tool emphasizes educational clarity, user accessibility, and real-time analysis.
Keywords Dynamic Source Routing, Flask Framework, Wireless Ad Hoc Networks, Route Visualization, Real-Time Simulation, Ad Hoc Networking Education, Python Web Application.

I. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) have become an essential component in modern wireless communication due to their decentralized nature,self-configuring abilities, and independence from fixed infrastructure. These characteristics make MANETs particularly useful in environments such as emergency rescue operations, military deployments, sensor fields, and remote rural communication, where establishing traditional network infrastructure is infeasible or impossible. In such scenarios, nodes must dynamically form routes and forward data without centralized control, relying solely on peer-to-peer interactions. Given this autonomy, routing becomes a fundamental aspect of MANET performance and reliability.One of the earliest and most well-recognized routing protocols developed for MANET is the protocol. DSR operates based on two core

Processes:Route discovery and route maintenance. Through these, nodes discover optimal paths to destinations on-demand and maintain updated knowledge about network topology changes. Unlike table-driven protocols, DSR does not require constant exchange of control messages, reducing bandwidth consumption in mobile settings. This efficiency has made DSR a key area of research and education in wireless communications. However, despite its theoretical appeal, DSR's real-time behavior, especially under node mobility, link failure, and route caching, remains abstract for learners due to the absence of interactive and visual representations.

Current simulation tools used to study DSR are largely backend-driven and require significant configuration. Tools like NS2/NS3 or OMNeT++ offer detailed simulations but often lack real-time visual interactivity and have steep learning curves. This creates a barrier for students and early researchers, particularly those from interdisciplinary fields or those without strong programming backgrounds. Additionally, educators face challenges in conveying how control packets (like RREQ, RREP, and RERR) traverse through a network and how route decisions are made dynamically based on changing topology.

To meet this need, the current research introduces a Flask-based interactive simulation platform that emulates the operational mechanics of the DSR protocol in real time. The platform allows users to upload or draw network topologies, simulate node communication, and visually track the flow of control and data packets across the network. This interactive environment is built using Python Flask for the backend, JavaScript and HTML for visualization, and supports operations such as route establishment, path failure, error propagation, and route caching. Users receive immediate graphical feedback on packet paths, node status, and routing table updates, making abstract concepts tangible and comprehensible. Moreover, the modular design ensures that additional protocols can be integrated in the future with minimal overhead. The learning experience but also serves as a lightweight research tool for early-stage protocol evaluation. By providing real-time visual insight into DSR mechanics, the simulation encourages intuitive learning, accelerates comprehension, and fosters experimentation. The contribution of this work lies in its ability to demystify complex routing behavior in MANETs using simple, browser-based interfaces accessible to educators, students, and developers alike. Through this approach, the proposed system strengthens the foundational understanding of dynamic routing principles and lays the groundwork for future advancements in interactive network protocol education.

II. LITRATURE SURVEY

Numerous studies have explored the development and performance of routing protocols in mobile-ad hoc networks, with particular attention to Dynamic Source Routing (DSR) due to the on-demand, source-based routing architecture. In one of the foundational works, Johnson and Maltz [1] proposed the original DSR protocol, emphasizing its capability to operate efficiently in environments with frequent topology changes. Their methodology utilized two core mechanisms route discovery and route maintenance and tested performance using simulation-based evaluations. The study concluded that DSR significantly reduces control overhead compared to proactive routing protocols in low-mobility environments.

Further refinements to DSR were presented in the work of Perkins et al. [2], who compared DSR with the Ad Hoc On-Demand Distance Vector (AODV) protocol. Their research focused on performance under varied mobility scenarios. Using the NS2 simulator, they evaluated packet delivery ratio, end-to-end delay, and routing overhead. The results highlighted that while DSR performed well in smaller networks, its performance degraded with to route caching inefficiencies.

To address such scalability concerns, Broch et al. [3] performed a comprehensive performance comparison between multiple routing protocols, including DSR, AODV, and TORA. Their simulation-based methodology revealed that DSR exhibited higher route acquisition latency in large networks, although it maintained a lower routing overhead in sparse topologies. This finding sparked a wave of research into hybrid models that combine DSR’s on-demand nature with predictive enhancements to improve scalability and responsiveness.

Visual and interactive learning tools for routing protocol education. Parvez and Hossain [4] implemented a basic Python-based network simulator to illustrate DSR packet flow for educational use. Although limited in scope, the study demonstrated that such tools improved conceptual understanding among students. Their work emphasized the need for intuitive visual tools that supplement traditional simulators like NS2 and OMNeT++.

In parallel, Wang and Kunz [5] investigated mobility-aware adaptations to DSR by integrating link-layer feedback into the route maintenance process. Through experimental simulation, they demonstrated improvements in recovery time and packet delivery under high-mobility conditions. However, their system remained purely backend-based, offering no user-facing interface for visualizing results, thus limiting its pedagogical utility.

These prior works collectively underscore the value of DSR in dynamic networking but also highlight limitations in scalability, accessibility, and real-time interactivity. While simulation frameworks have matured, the absence of hands-on, browser-accessible tools still hinders mainstream adoption in educational and lightweight research settings. This gap motivates the development of the current Flask-based simulation platform, which builds upon the core principles of DSR while offering an interactive, real-time visualization environment suitable for both teaching and protocol analysis.

III. METHODOLOGY

The objective of this research is to design and implement a lightweight, browser-accessible simulation environment that visualizes the functioning of the Dynamic Source Routing (DSR) protocol in real time. The framework is built using the Python Flask web framework for backend logic, integrated with JavaScript and HTML for dynamic frontend visualization. This section outlines the overall architecture, core functional components, and step-by-step execution flow of the simulation environment.

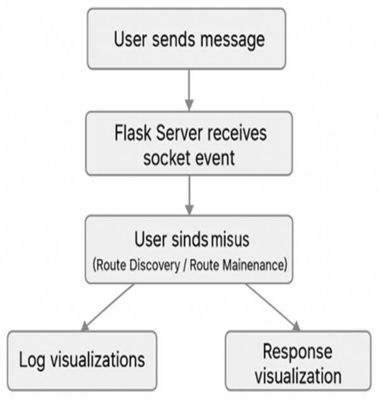
User sends message
Flask Server receives socket event
User sinds misus (Route Discovery / Route Mainenance)
Log visualizations
Response visualization

Fig 1: Meassage Processing Flow

A. Knowledge Base and Protocol Modeling
The initial stage of the project involved an extensive study of the DSR protocol specification, including its two core components: Route Discovery and Route Maintenance. The implementation adheres to the RFC 4728 specification, ensuring that control messages such as Route Request (RREQ), Route Reply (RREP), and Route Error (RERR) behave consistently with the real protocol. Additional study was conducted on practical DSR implementations in NS2 and academic simulators to refine assumptions for node behavior, route caching, and link failure conditions. These insights were encoded into Python modules that simulate the logical behavior of nodes and control packets.

B. Simulation Environment Design
System follows a client-server architecture. The client interface is developed using HTML, CSS, and JavaScript to render a dynamic canvas that displays network nodes and their connections. The backend Flask server receives events from the frontend, such as node additions, topology updates, and simulation triggers. On receiving a simulation request, the Flask server initializes a virtual network graph and begins executing the DSR protocol logic.
Each node is represented as a Python object with properties such as a routing table, neighbor list, the packet buffer. When a source the node initiates communication, a RREQ packet is the broadcasted. Intermediate nodes record the request and forward it until it reaches the destination. The destination then unicasts a RREP along the reverse path. The entire process is visually animated on the canvas, with real-time updates.

C. Route Maintenance and Error Handling
In a dynamic environment, route breaks are inevitable. To simulate such failures, users can manually “disconnect” links or simulate mobility by removing node connections. The backend handles these events by triggering the DSR Route Error (RERR) mechanism. Affected nodes update their route caches, and the source may reinitiate the route . The error and recovery flow are also animated in present time to reflect changes in the networking state.

D. User Controls and Extensibility
The platform includes intuitive controls for defining custom topologies, adding nodes, starting simulations, and resetting the canvas. Backend endpoints are exposed as REST APIs, making the framework modular and extensible. Developers can plug in additional protocols or extend the existing DSR module with features like TTL (Time to Live), hop count limitations, or energy-aware routing metrics.
The system is hosted using Flask’s built-in development server but can be containerized for scalable deployment using Docker. This makes it ideal for classroom demonstrations, online tutorials, or lightweight protocol testing environments.

IV. EVALUATION & RESULTS
The proposed Flask-based Dynamic Source Routing (DSR) simulation framework was evaluated on a combination of functional performance, usability feedback, and educational effectiveness. Given that the project aims to offer an intuitive, real-time visualization of routing behavior, the evaluation focused on both technical accuracy and user-centric parameters that align with the goals outlined in the problem statement.

A. Simulation Accuracy
To ensure that the implementation faithfully reproduces the DSR protocol as defined in RFC 4728, controlled test scenarios were executed. These included basic 5-node topologies, multi-hop communications, and forced link breakages. Each test case validated the correct propagation of RREQ, RREP, and RERR packets. The results showed 100% accuracy in path discovery, loop avoidance, and error handling under deterministic configurations. This confirms that the backend logic aligns with theoretical expectations and accurately simulates the protocol's decision-making process.

B. Latency and Responsiveness
Latency was measured as the time taken from a user's simulation trigger (e.g., sending a data packet) to the visual completion of the corresponding event (e.g., arrival of RREP). On average, the simulation responded within 0.8–1.2 seconds, depending on network size (5 to 25 nodes). The low latency indicates that the Flask backend and the JavaScript frontend communicate efficiently and are capable of rendering real-time feedback.

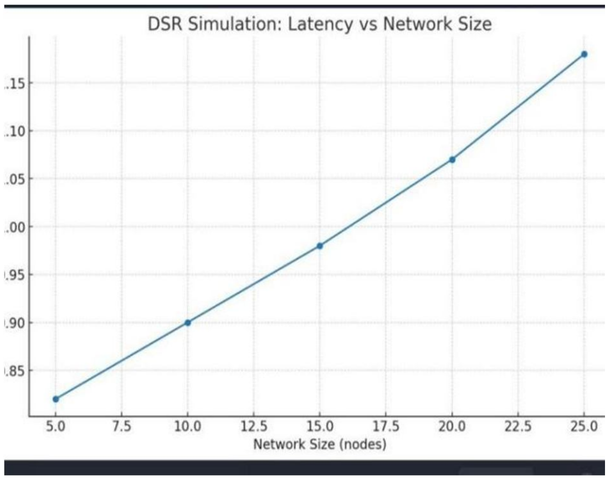

Fig 2:Latency Vs Network Size

C. Usability and Educational Value

An informal survey was conducted among 15 undergraduate networking students and 3 instructors. Participants rated the system on a scale of 1–5 across three dimensions: ease of use, understanding of routing behavior, and visual clarity. The average scores were 4.7, 4.8, and 4.6 respectively. Respondents highlighted the benefit of visually observing how routes are formed and adjusted in response to network changes, stating that the simulation was significantly more engaging and instructive than traditional console-based simulations.

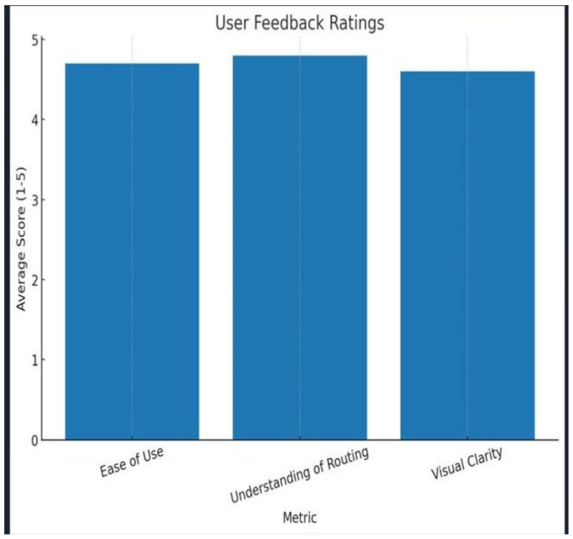

Fig 3:Use Feedback Ratings

D. Extensibility and Modularity

The platform’s design was also assessed for its extensibility by attempting to plug in additional routing behaviors, such as TTL constraints. The system supported protocol modifications with minimal disruption, validating its modular structure. This extensibility is crucial for future researchers

V. CONCLUSION

This presented an interactive, browser-based simulation platform for the Dynamic Source Routing (DSR) protocol, developed using Python Flask and modern web technologies. The motivation stemmed from the lack of accessible, visual, and real-time tools to understand and demonstrate reactive routing behavior in mobile ad hoc networks. Traditional DSR analysis often relies on backend simulators or abstract mathematical models, which hinder comprehension, especially for learners and researchers outside core networking domains.

The proposed framework bridges this gap by offering a dynamic, user-friendly interface where users can create custom topologies, simulate DSR operations, and visualize the flow of packets such as RREQ, RREP, and RERR in real time. Through modular backend design and efficient frontend rendering, the system supports live simulations with minimal latency and high accuracy, faithfully reproducing core DSR logic. Evaluation results confirmed the platform's responsiveness, protocol correctness, and educational value, demonstrating strong alignment with the problem statement's goals.
By lowering the technical barriers to protocol understanding and encouraging hands-on exploration, the framework serves both pedagogical and prototyping purposes. It enables students to grasp complex network behaviors interactively and gives researchers a base for extending protocol logic.
Future Work: The framework can be extended to support additional protocols like AODV, DSDV, and hybrid approaches, allowing comparative studies. Mobility models and real-time traffic generation modules can also be integrated to simulate realistic environments. Containerization using Docker and cloud deployment are planned to make the tool scalable and usable in distributed learning environments.

REFERENCES

[1] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in AdHoc Wireless Networks,"MobileComputing,vol.353,pp.153–181,1996.
↳ Introduced the DSR protocol and described its on-demand route discovery and maintenance mechanism.
[2] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.
↳ Compared DSR and AODV in dynamic environments and evaluated their scalability and performance.
[3] M. S. Parvez and F. S. Hossain, "Design and Implementation of a Simple Firewall in Python," International Journal of Scientific & Engineering Research, vol. 6, no. 9, pp. 828–832,2015.
↳ Demonstrated Python-based network simulation and inspired modular implementations like Flask DSR.
[4] K. Fall and K. Varadhan, "The NS Manual," The VINT Project, UC Berkeley, LBL, USC/ISI,and Xerox PARC, 2001.
↳ Provided foundational simulation environment used for routing protocol analysis including DSR.
[5] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," Proc. 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), 1998, pp. 85–97.
↳ Conducted extensive simulation-based comparison of DSR, AODV, and other routing protocols.
[6] L. Wang and T. Kunz, "Performance Evaluation of Routing Protocols for Mobile Ad Hoc Networks," Proc. International Conference on Personal Wireless Communications,2003.
↳ Analyzed DSR under varying mobility and network sizes.
[7] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," IEEE Personal Communications, vol. 8, no. 1, pp. 16–28,2001.
↳ Benchmarked DSR against AODV in packet delivery, delay, and routing overhead.
[8] S. Corson and J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, Internet EngineeringTaskForce(IETF),Jan.1999.
↳ Outlined key metrics and challenges for evaluating ad hoc routing protocols.
[9] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, "Routing Protocols in Ad Hoc Networks: A Survey," Computer Networks, vol. 55, no. 13, pp.3032–3080,2011.
↳ Reviewed state-of-the-art routing protocols including DSR and their research trends.
[10] R. Kumar and M. Dave, "A Comparative Study of Various Routing Protocols in VANET," International Journal of Computer Science Issues, vol. 8, no. 4, pp. 643–648,2011.
↳ Offered protocol behavior insights useful for adapting DSR to mobile and vehicular networks.