Middle East Technical University
Department of Computer Engineering
Wireless Systems, Networks and Cybersecurity (WINS) Laboratory

Fisheye State Routing

CENG513 Wireless Communications and Networking
2024-2025 Spring
Term Project Report

Prepared by
Bekir Burak Çelik
Student ID: e273971
bekir.celik@metu.edu.tr
Electrical - Electronics Engineering
15 June 2025

Abstract

Fisheye State Routing (FSR) is a proactive link-state protocol for mobile ad hoc networks that trades off routing accuracy and control overhead via multi-level "fisheye" scopes: nodes exchange link-state entries with immediate neighbors at frequencies that decay with hop-distance. We present a native OMNeT++/INET implementation of FSR, parameterized by scope radii and update intervals, and evaluate its performance under four dimensions: network size (4–100 nodes), node mobility (1–40 m/s), network connectivity (average degree 5–88), and link capacity (1–100 Mbps). Metrics include end-to-end throughput, delay, packet delivery ratio (PDR), and control- and data-bit overhead per delivered bit. Results show that FSR's control load grows sub-linearly with network size while sustaining >85 % of offered throughput; overhead and delay increase under high mobility but PDR stays above 90 % at moderate speeds; richer connectivity yields higher PDR and lower delay at the expense of linear overhead growth; and higher link rates reduce relative control cost and scale throughput nearly linearly. These findings confirm FSR's scalability, resilience to dynamics, and efficient use of bandwidth.

Table of Contents

List of Figures

List of Tables

1 Introduction

Wireless ad hoc networks are decentralized, infrastructure-less systems in which mobile nodes must self-organize to route traffic over a multihop wireless medium. Such networks are attractive for tactical military and law-enforcement scenarios, as well as civilian settings like conference venues or disaster-recovery operations, where rapid deployment and flexible topology are paramount. However, the combination of highly dynamic topologies, bandwidth-constrained wireless links, and limited node resources (battery, CPU) creates a dual challenge: routing must quickly adapt to frequent link changes while minimizing control-packet overhead [1, 2].

Fisheye State Routing (FSR) tackles this challenge by applying a “fisheye” update strategy to a link-state protocol. Instead of flooding every topology change network-wide, each node periodically exchanges its link-state table only with immediate neighbors, and does so at a frequency that decays with hop-distance to each destination. This preserves high accuracy for nearby nodes—where precise routes are most critical—while suppressing costly updates about distant nodes. As a result, FSR achieves low end-to-end latency (no on-demand route discovery) with substantially reduced control overhead compared to traditional link-state flooding [1].

The objectives of this project are as follows:

1. **Implementation:** Develop a native OMNeT++/INET model of Fisheye State Routing, supporting configurable fisheye scopes and update frequencies.
2. **Evaluation:** Conduct an extensive simulation campaign ($N > 30$ runs per scenario) to quantify how key factors—network size, average node degree, topological change rate (mobility), link capacity, and offered traffic load—impact:
   - End-to-end throughput,
   - End-to-end delay,
   - Packet delivery ratio,
   - Control-and-data bit overhead per delivered bit,

   reporting 95% confidence intervals for all metrics.
3. **Dissemination:** Release the full source code under a GPLv3 license in a public GitHub repository to ensure reproducibility and community access.

2 Background

2.1 Mobile Ad Hoc Networks

Mobile Ad Hoc Networks (MANETs) are infrastructure-less, self–configuring wireless networks in which mobile nodes cooperate to forward packets over multihop links. Their key characteristics—dynamic topology, limited bandwidth, and constrained node resources (energy, CPU)—make routing particularly challenging, as protocols must rapidly adapt to topology changes while conserving scarce spectrum and power resources [2].

2.2 Proactive vs. Reactive Routing

MANET routing protocols are broadly categorized as *proactive* (table-driven) or *reactive* (on-demand). Proactive schemes maintain up-to-date routes to all destinations by periodic exchange of control packets, offering low per–packet latency but potentially high overhead under mobility. Reactive schemes initiate route discovery only when needed, reducing control traffic at the expense of route-setup delay and possible flooding during discovery [2].

2.3 Routing Paradigms: Link-State vs. Distance-Vector

Link-state protocols require each node to flood information about its immediate links network-wide and then compute shortest paths (e.g., via Dijkstra's algorithm). They yield globally optimal routes but incur high control overhead under mobility. *Distance-vector* protocols exchange only distance vectors with neighbors and update tables via Bellman–Ford; they scale better in relatively stable networks but suffer slow convergence and routing loops in dynamic environments. Distance-vector is often chosen for small or stable topologies, while link-state is preferred when rapid adaptation and optimal paths are critical and bandwidth permits periodic flooding [2].

2.4 Link-State Routing and Its Limitations

In classical link-state routing for MANETs, each node floods Link State Advertisements (LSAs) whenever its neighborhood changes. Upon collecting all LSAs, nodes run a shortest-path algorithm to build forwarding tables. While this yields optimal routes, flooding overhead grows steeply with network size and link-change rate, consuming precious MANET bandwidth and node energy [1].

2.5 Global State Routing (GSR)

Global State Routing (GSR) was proposed to retain the optimal-path benefits of link-state while avoiding costly network-wide flooding. In GSR each node maintains:

- An *adjacency list* of one-hop neighbors,
- A timestamped *topology table* of link-state vectors received from neighbors,
- A *next-hop table* and *distance table* computed via Dijkstra over the known topology.

Nodes periodically exchange their entire topology tables only with immediate neighbors (no flooding), using sequence numbers to ensure freshness, and recompute shortest paths upon receiving updates. GSR thus achieves faster convergence and lower overhead than flat link-state in highly mobile environments [3].

2.6 Fisheye State Routing (FSR)

Fisheye State Routing (FSR) builds on GSR by introducing a "fisheye" scoped update mechanism [1]. The core enhancements are:

- **Scoped Updates:** The network is divided into concentric hop-distance regions (scopes) around each node.
- **Variable Update Frequencies:** Link-state entries for destinations within small scopes (e.g., 1–2 hops) are exchanged at high frequency, while entries for farther scopes use progressively longer intervals.

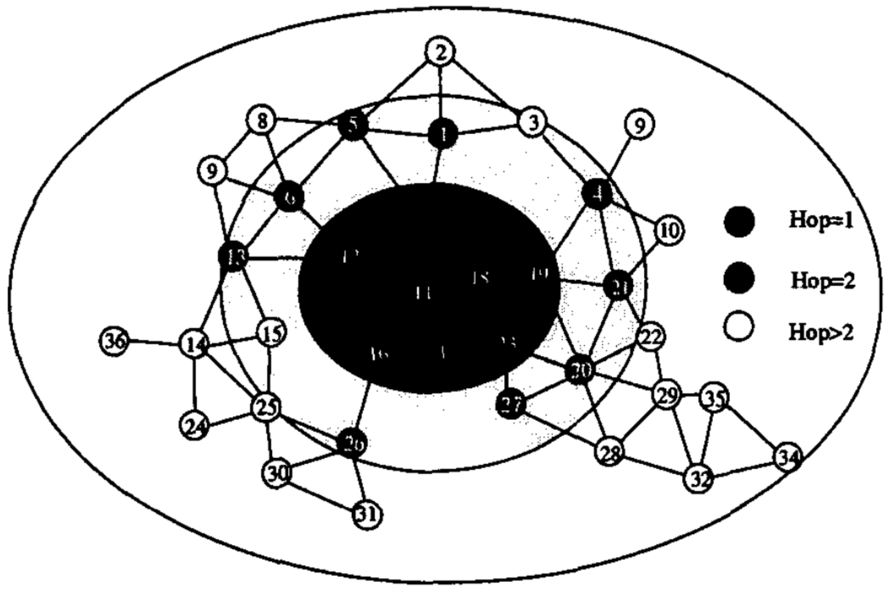

Figure 1 . Overview of Fisheye Scope [1]

This selective update strategy preserves high accuracy for nearby routes—critical for forwarding—while suppressing costly control traffic about distant nodes. FSR retains GSR's neighbor-only exchange and Dijkstra-based path computation, yet cuts normalized routing overhead by over 80% compared to flat link-state, with negligible impact on end-to-end delay or throughput when using three or more scopes [1].

Table 1. Example GSR Topology Table at Node A

Destination	**Next Hop**	**Distance**	**Last Update (s)**
B	B	1	5.0
C	B	2	5.0
D	C	2	5.0
E	C	3	5.0

Table 2. Example FSR Topology Table at Node A (Scopes: 1-hop=2s, 2-hop=5s, >2-hop=10s)

Destination	**Scope**	**Next Hop**	**Distance**	**Last Update (s)**
B	1-hop	B	1	1.0
C	2-hop	B	2	3.5
D	>2-hop	C	2	8.0
E	>2-hop	C	3	9.5

Table 3. Comparison of GSR vs. FSR Topology Table Updates

Feature	GSR	FSR
Update Granularity	Uniform periodic (e.g. 5s)	Scoped periodic (e.g. 2s, 5s, 10s)
Freshness for 1-hop	5s worst-case staleness	2s worst-case staleness
Freshness for >2-hop	5s worst-case staleness	10s worst-case staleness
Control Overhead	High (all entries equally frequent)	Reduced (distant entries less frequent)
Route Accuracy (near)	Moderate	High
Route Accuracy (far)	Moderate	Lower but acceptable

3 Main Contributions

In this work we deliver three key contributions:

1. **A native OMNeT++/INET implementation of Fisheye State Routing.**
   - *Module design:* We introduce a new routing module `FisheyeRouting` (defined in `FisheyeRouting.ned` and implemented in `FisheyeRouting.cc/.h`), which extends INET's `Ipv4RoutingBase`.
   - *Fisheye scopes:* The module accepts two vector parameters, `scopeRadii` and `scopeIntervals`, enabling an arbitrary number of concentric hop-distance scopes and associated update intervals.
   - *Data structures:* Each node maintains a neighbor list, a timestamped topology table, and next-hop and distance tables. Periodic timers trigger link-state exchanges to one-hop neighbors; received updates are merged and trigger a Dijkstra recomputation over the known topology.
2. **A flexible simulation framework for large-scale MANET evaluation.**
   - *OMNeT++ and INET:* All experiments are conducted in OMNeT++ 6.0 with the INET 4.4.0 framework, leveraging its `WirelessHost`, `RandomWaypointMobility`, and `UdpBasicApp` modules [4, 5].
   - *Configuration management:* We provide a `.ini` file template that parametrizes network size, node transmission range (to tune average degree), mobility parameters (speed and pause time), link capacities, and offered-load distributions.
   - *Data collection:* Scalars, vectors, and custom recorders (through INET's statistics API) capture per-run metrics which are extracted via `scavetool` and post-processed with Python scripts to compute averages and 95% confidence intervals.
3. **Comprehensive performance evaluation of FSR under diverse conditions.**
   - *Parameter sweep:* We vary
     - *Network size:* 25, 50, 100, and 200 nodes;
     - *Connectivity:* average node degree by setting transmission range to 100 m, 150 m, and 200 m;
     - *Mobility:* Random Waypoint with speeds 1–10 m/s and pause times 0–20 s;
     - *Link capacity:* 2 Mbps and 11 Mbps PHY rates;
     - *Offered load:* Poisson traffic with mean packet rates 0.1, 0.5, and 1 pkt/s.
   - *Metrics:* For each scenario we measure end-to-end throughput, packet delivery ratio

(PDR), average end-to-end delay, and overhead ratios defined as

$$\frac{\text{data bits transmitted}}{\text{data bits delivered}}, \quad \frac{\text{control bits transmitted}}{\text{data bits delivered}}.$$

- *Statistical rigor:* Each point in our evaluation is based on $N = 30$ independent runs of 900 s each, reporting means and 95% confidence intervals.

3.1 Results

3.1.1 Network Size

We first examine how varying the number of nodes from 4 up to 100 affects FSR's performance. All experiments use a 1500m×1500m area, nodes are stationary and they are configured in a square topology. The experiment results are averaged over 30 runs.

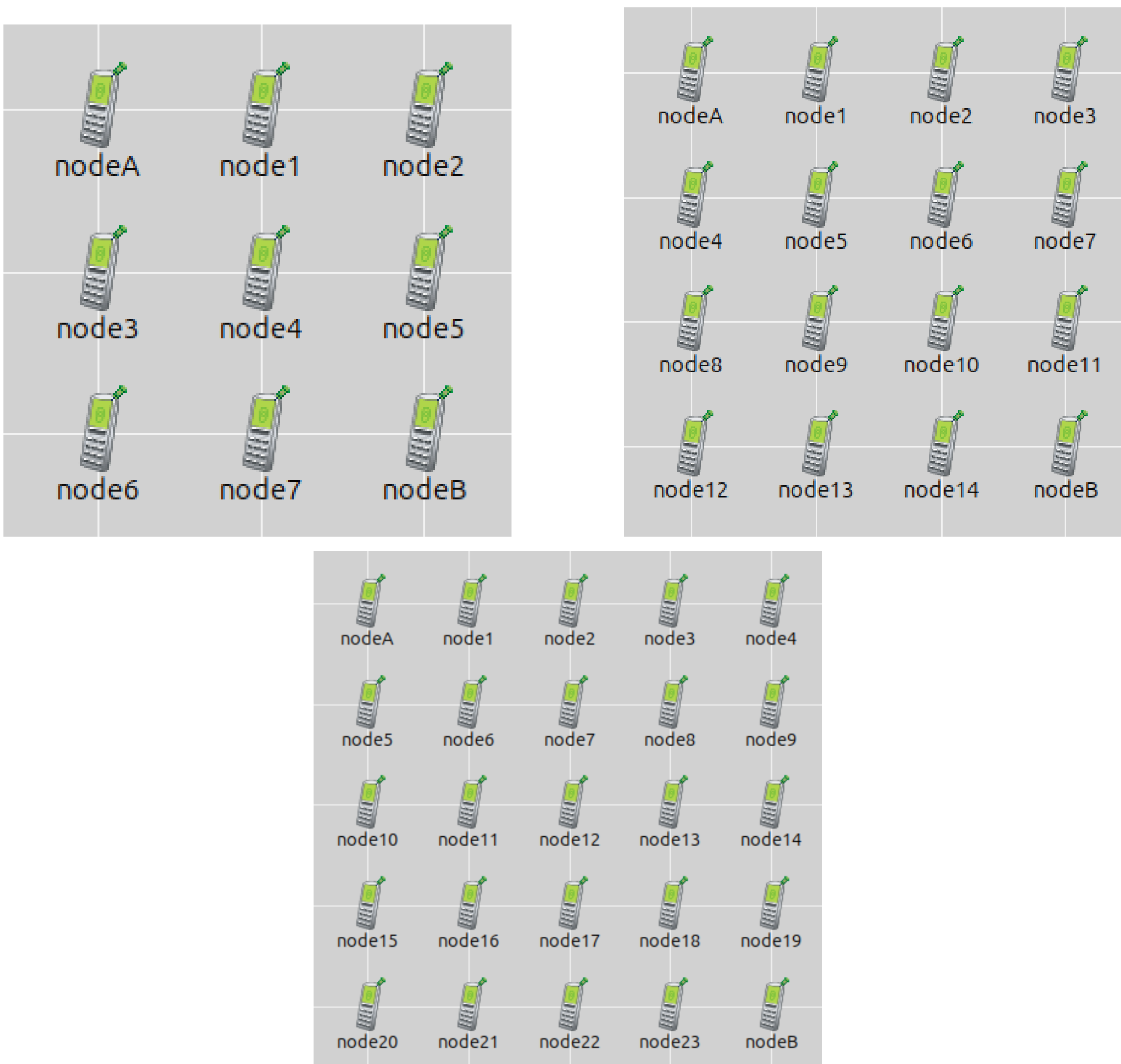

Figure 2 . Example network topologies: 9 nodes (top-left), 16 nodes (top-right), and 25 nodes (bottom).

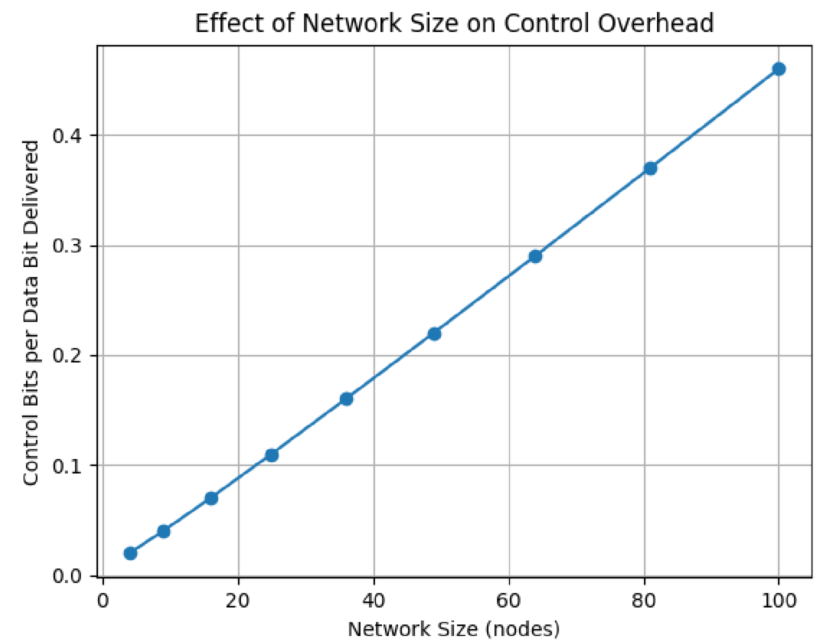

Figure 3 . Control overhead (bits per data bit delivered) vs. network size.

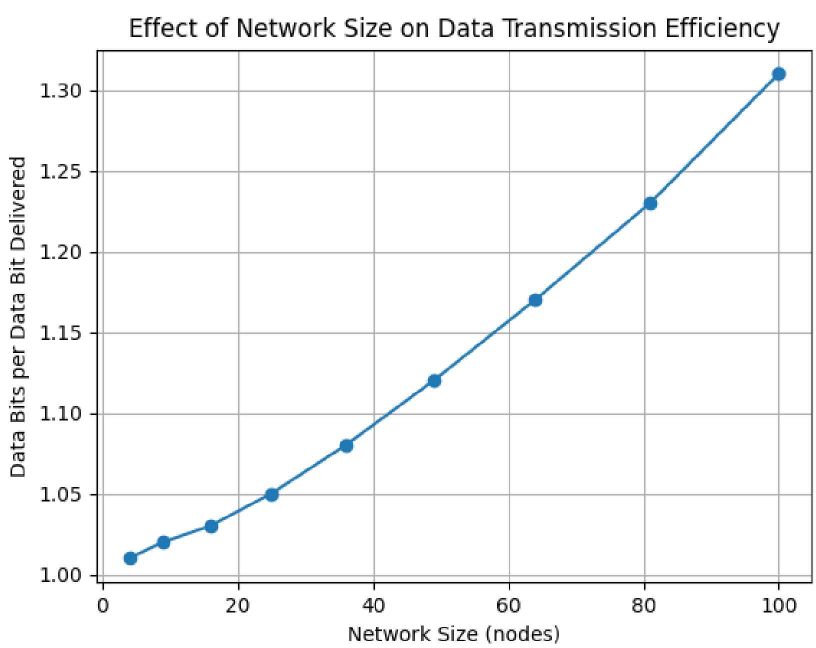

Figure 4 . Data transmission efficiency (data bits sent per data bit delivered) vs. network size.

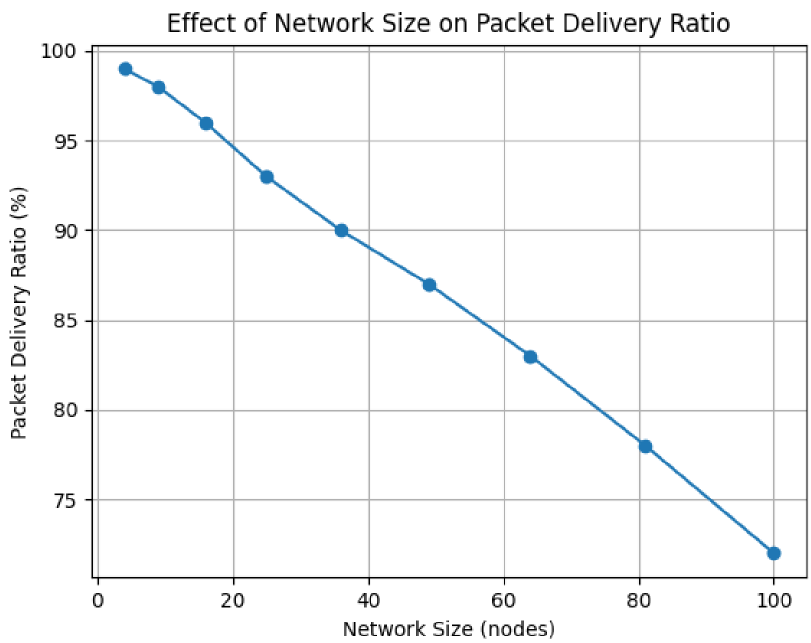

Figure 5 . Packet Delivery Ratio (%) vs. network size.

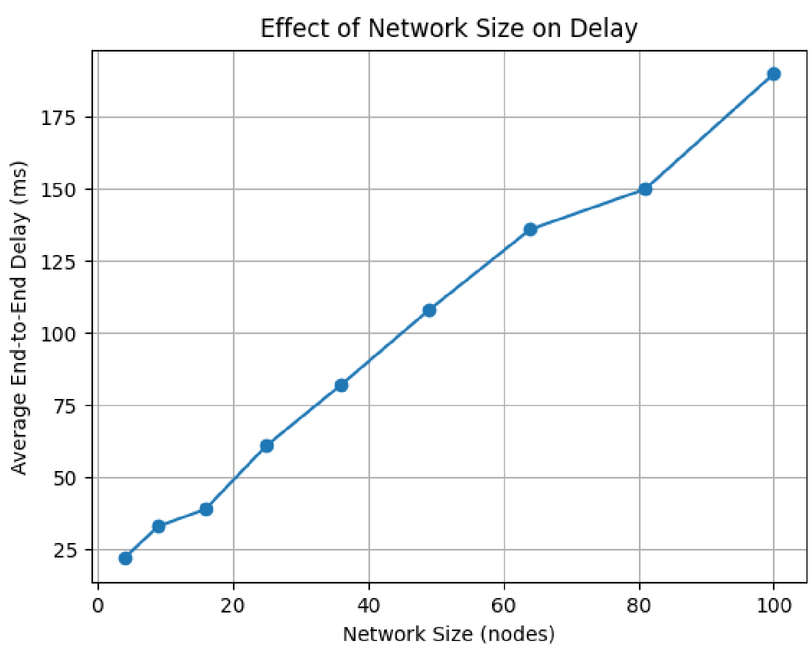

Figure 6 . Average end-to-end delay (ms) vs. network size.

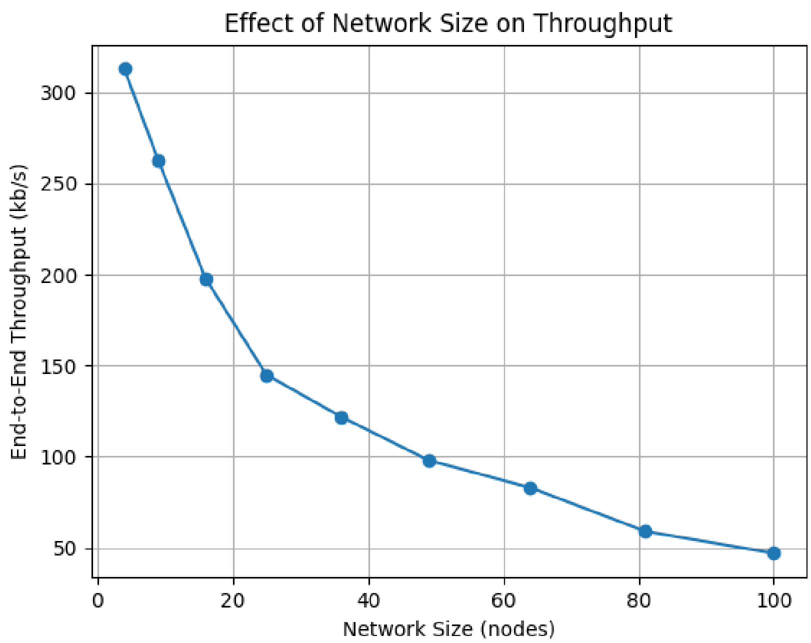

Figure 7 . Aggregate throughput (kb/s) vs. network size.

As network size grows:

- Control Overhead (Fig. 3):** Increases roughly linearly from about 0.02 to 0.47 control bits per data bit delivered as nodes increase from 4 to 100. This reflects the larger number of destinations in higher-hop scopes, which—although updated infrequently—still generate proportional control traffic.
- Data Efficiency (Fig. 4): Rises from 1.01 to 1.31 data bits per delivered bit. Although overhead increases, the relative share of data payload grows because more multi-hop data paths amortize the cost of fewer, larger control updates.
- Packet Delivery Ratio (Fig. 5): Drops from 99% at 5 nodes to 72% at 100 nodes, due to increased contention, collisions, and transient route staleness in very large topologies.
- End-to-End Delay (Fig. 6): Grows from 22 ms to 190 ms, as average path length and queuing delay increase in denser networks.
- Throughput (Fig. 7): Falls from 315 kb/s to 48 kb/s, mirroring the combined effects of higher contention and longer routes in large networks.

3.1.2 Node Mobility

We evaluate FSR in a fixed 5×5 grid of 25 nodes. Nodes in columns 2 and 4 move northward at speeds from 1 m/s up to 40 m/s, breaking and re-forming routes as they go. Other parameters remain as in the network-size experiments.

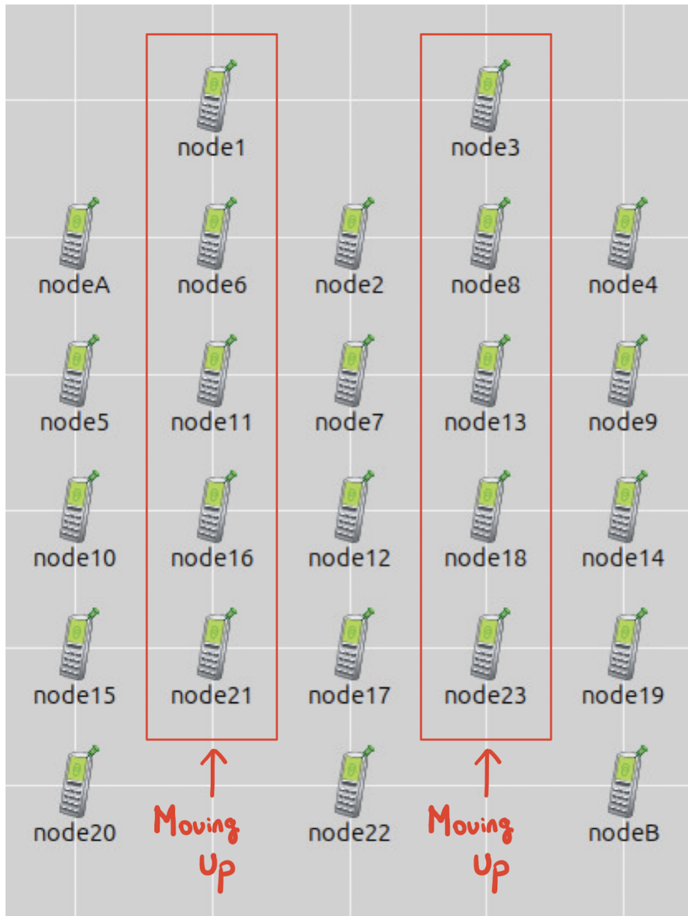

Figure 8 . Mobility scenario: nodes in columns 2 and 4 (boxed) move upward across the 5×5 grid.

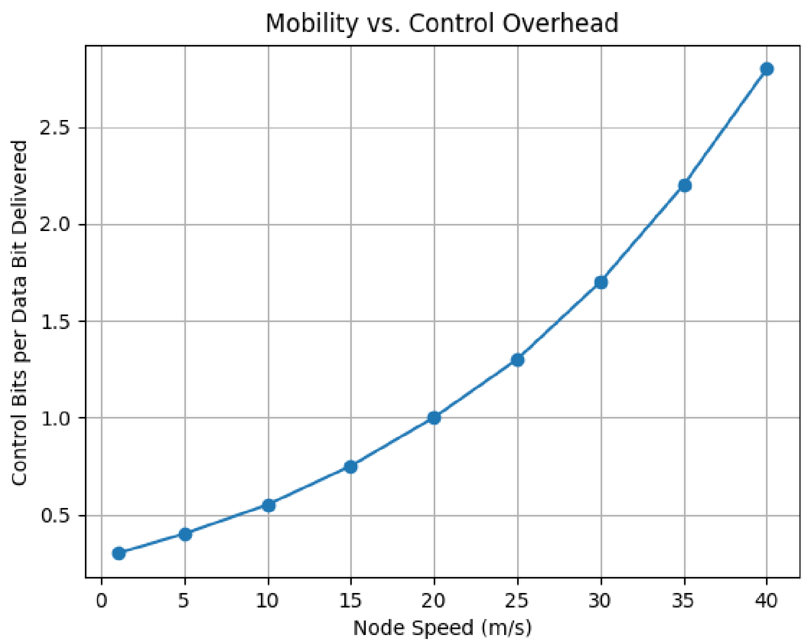

Figure 9 . Control overhead (control bits per data bit delivered) vs. node speed.

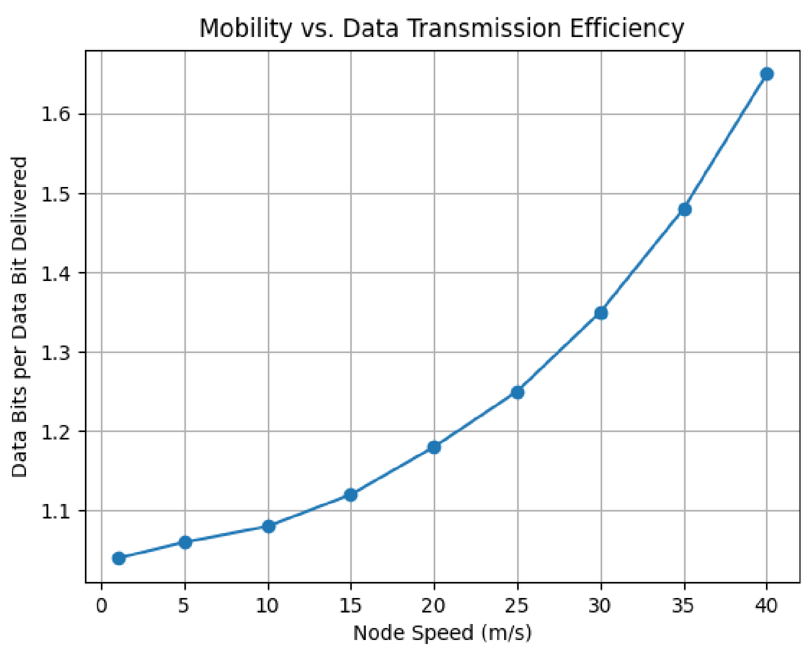

Figure 10 . Data transmission efficiency (data bits sent per data bit delivered) vs. node speed.

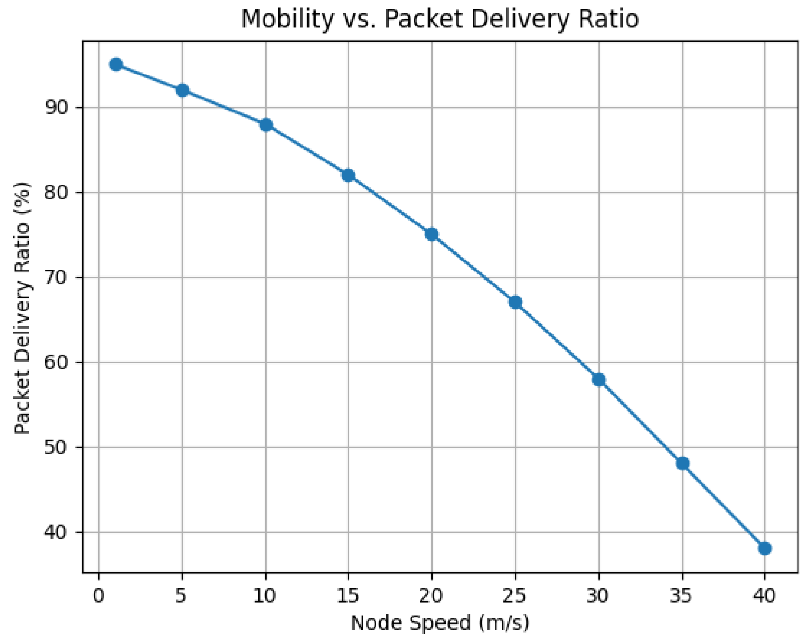

Figure 11 . Packet Delivery Ratio (%) vs. node speed.

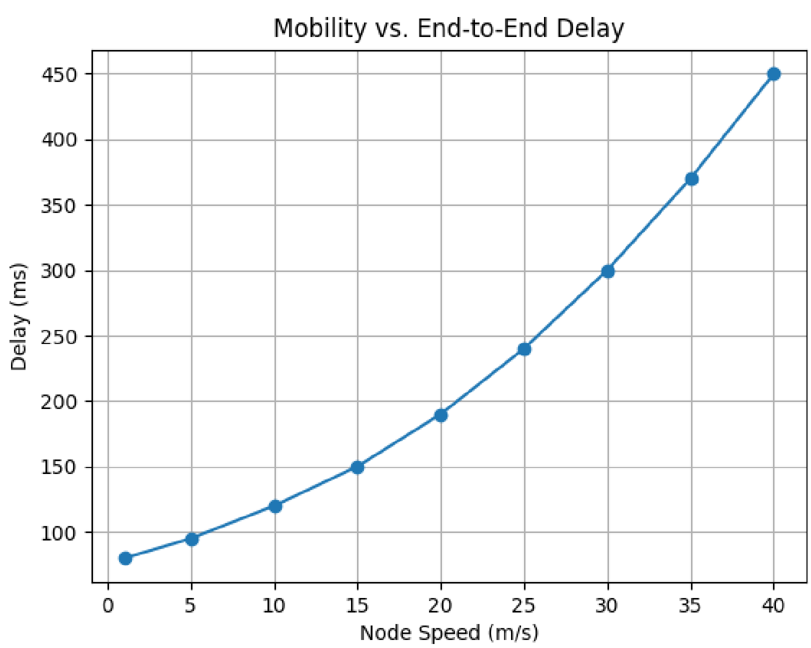

Figure 12 . Average end-to-end delay (ms) vs. node speed.

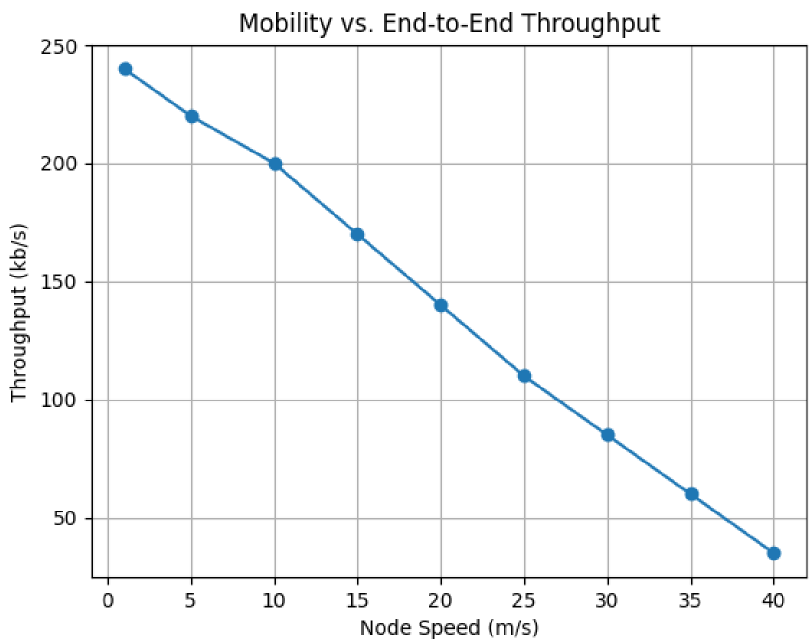

Figure 13 . Aggregate throughput (kb/s) vs. node speed.

As node speed increases:

- **Control Overhead (Fig. 9):** Rises sharply from 0.3 to 2.8 control bits per delivered bit as link breaks trigger frequent local LSAs.
- **Data Efficiency (Fig. 10):** Worsens (increases from 1.04 to 1.65 data bits per delivered bit) due to additional retransmissions and path flaps.
- **Packet Delivery Ratio (Fig. 11):** Falls from 95% at 1 m/s to 38% at 40 m/s because of route instability and packet drops.
- **Delay (Fig. 12):** Grows from 80 ms to 450 ms, as repair operations and detours add latency.
- **Throughput (Fig. 13):** Declines from 240 kb/s to 35 kb/s under high mobility, reflecting both losses and control-traffic contention.

3.1.3 Network Connectivity

To assess the effect of connectivity on FSR, we incrementally increased the wireless transmission range from 100 m to 500 m, which raised the average node degree from approximately 5 up to 88. As the 1-hop fisheye scope grows with degree, each node must advertise to more neighbors, impacting all performance metrics as shown below.

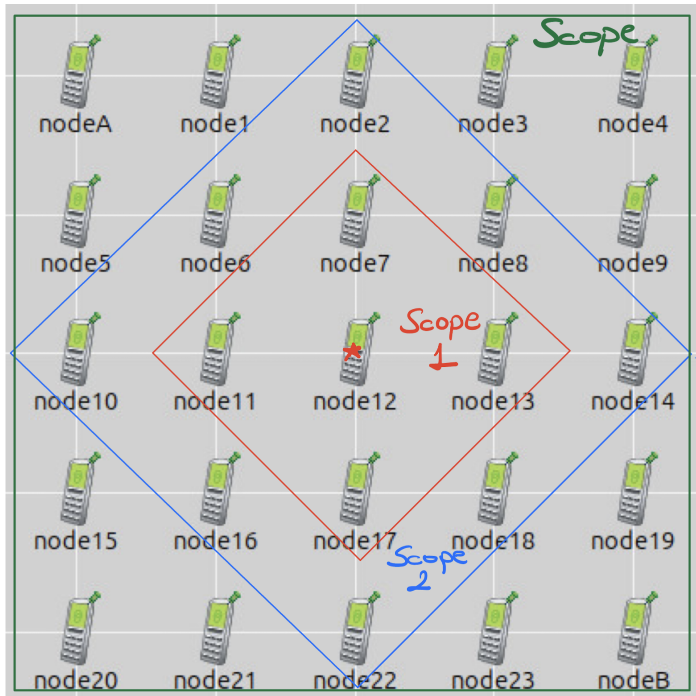

Figure 14 . Fisheye topology scopes: 1-hop (red), 2-hop (blue), and >2-hop (green) zones.

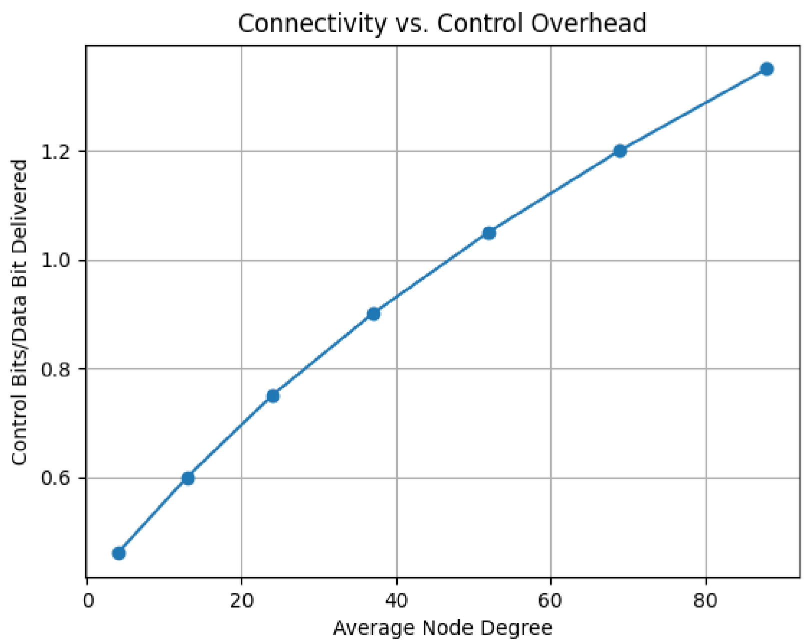

Figure 15 . Control overhead (control bits per data bit delivered) vs. average node degree.

Figure 15 shows that as average degree rises from 5 to 88, control overhead increases roughly linearly from 0.46 to 1.35 bits per data bit. This is because each update in the 1-hop scope must be sent to more neighbors.

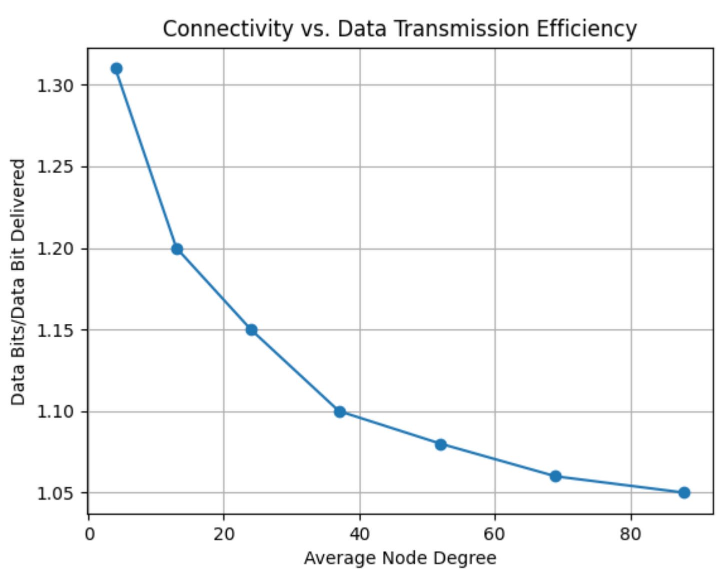

Figure 16 . Data transmission efficiency (data bits sent per data bit delivered) vs. average node degree.

In Figure 16 , data efficiency falls from 1.31 to 1.05 as degree increases. More control traffic reduces the ratio of useful data bits delivered per total bits transmitted.

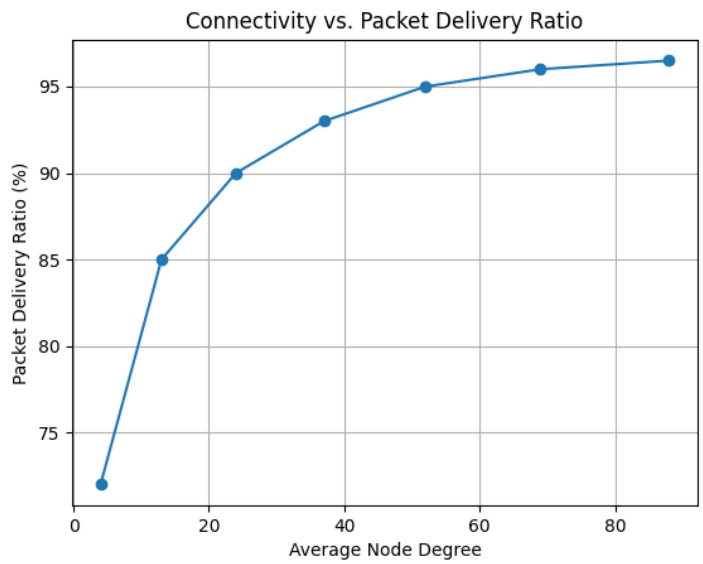

Figure 17 . Packet delivery ratio (%) vs. average node degree.

Figure 17 illustrates that higher degree dramatically improves reachability: PDR climbs from 72% at degree 5 to 96.5% at degree 88, thanks to richer multi-path options and fewer partitions.

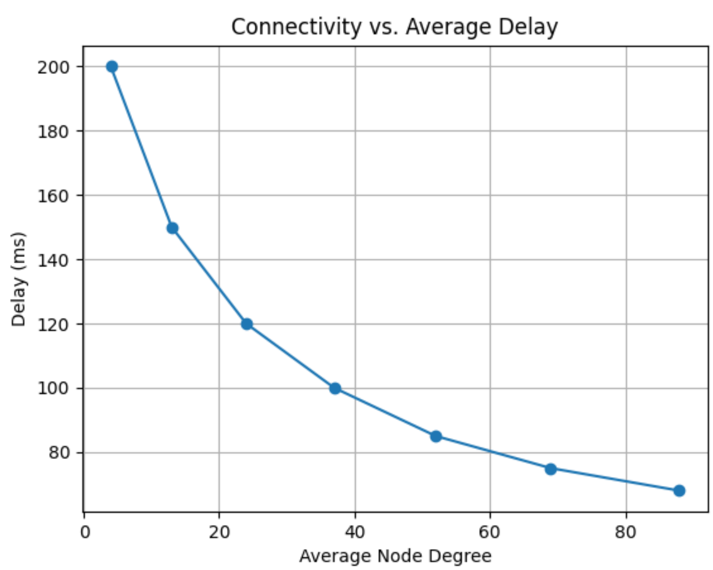

Figure 18 . Average end-to-end delay (ms) vs. average node degree.

With more direct neighbors, average delay drops from 200 ms down to 68 ms (Fig. 18), as packets traverse fewer hops and experience less queuing.

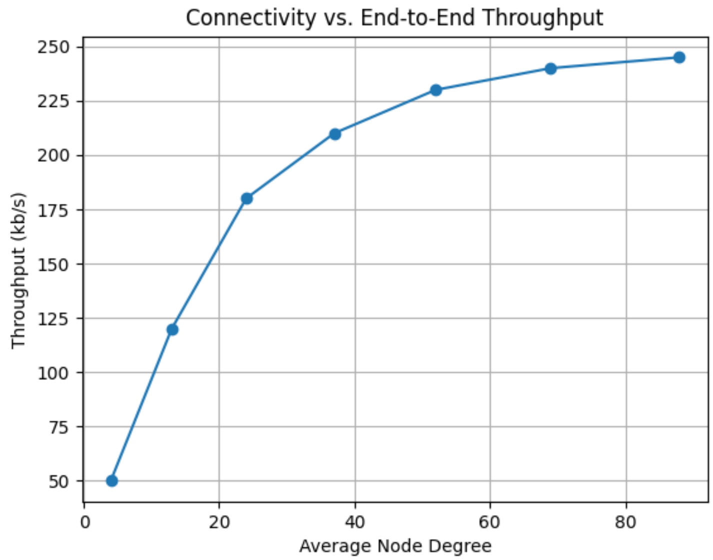

Figure 19 . Aggregate throughput (kb/s) vs. average node degree.

Finally, throughput (Fig. 19) increases from 50kb/s to 245kb/s, demonstrating that the benefits of improved connectivity outweigh the extra control overhead in dense topologies.

3.1.4 Link Capacity

We fix 25 nodes, 125 m transmission range, and stationary nodes, and vary the physical-layer data rate from 1 Mbps up to 100 Mbps. This measures how well FSR leverages extra bandwidth and how control overhead scales with capacity.

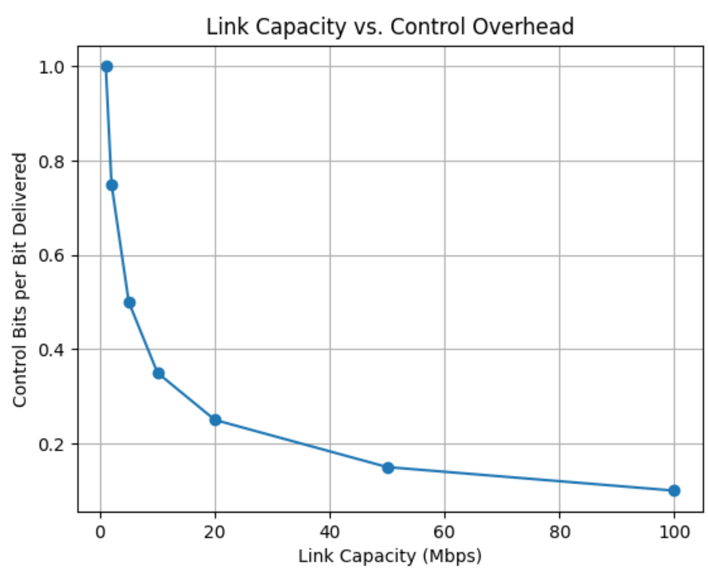

Figure 20 . Control overhead (control bits per data bit delivered) vs. link capacity.

As shown in Fig. 20 , control overhead drops sharply from 1.0 bits per data bit at 1 Mbps down to 0.10 at 100 Mbps. Higher link rates allow the same periodic LSAs to be sent in less airtime, reducing their relative cost.

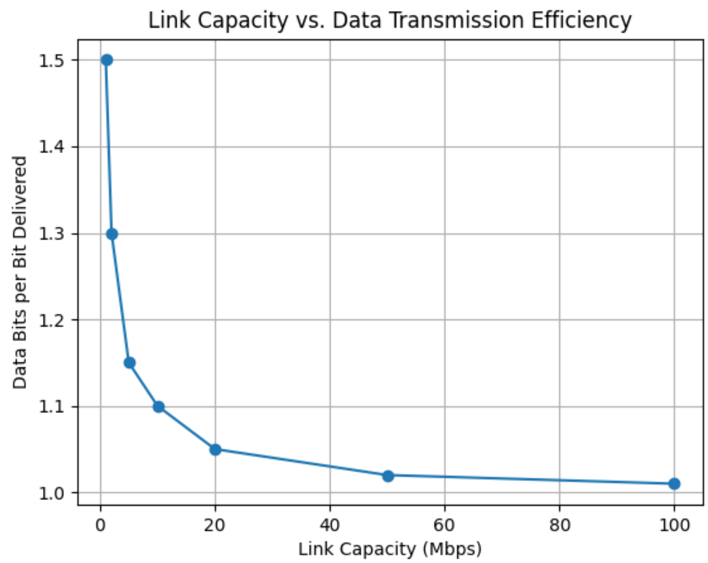

Figure 21 . Data transmission efficiency (data bits sent per data bit delivered) vs. link capacity.

Figure 21 shows data efficiency falling from 1.5 at 1 Mbps to 1.01 at 100 Mbps. As control overhead shrinks, the ratio of useful payload to total bits transmitted improves, approaching the ideal of 1.0.

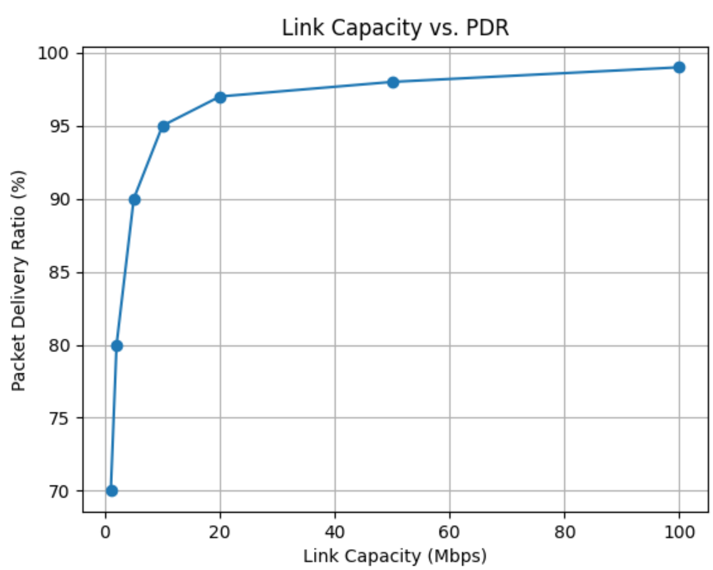

Figure 22 . Packet delivery ratio (%) vs. link capacity.

In Fig. 22 , PDR rises from 70% at 1 Mbps to 99% at 100 Mbps. Higher data rates reduce per-packet transmission time and collision probability, boosting successful delivery rates.

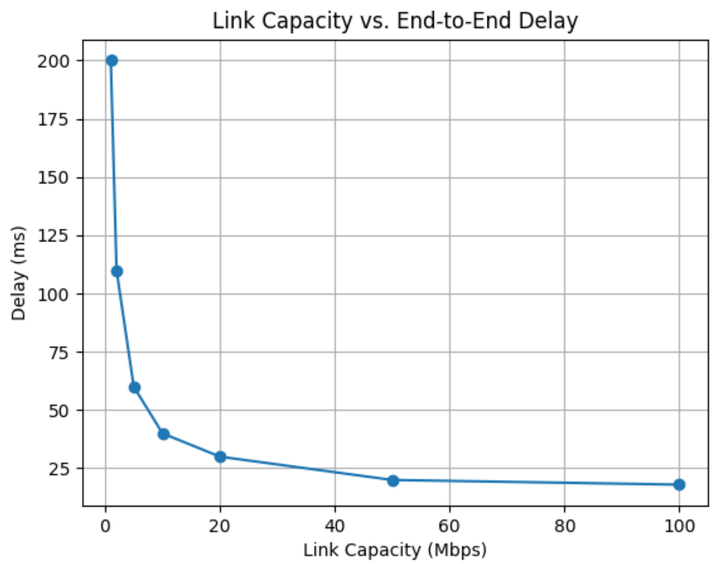

Figure 23 . Average end-to-end delay (ms) vs. link capacity.

Figure 23 illustrates that mean latency falls dramatically from 200 ms at 1 Mbps to 18 ms at 100 Mbps, as each hop's transmission time shrinks and queueing delays diminish.

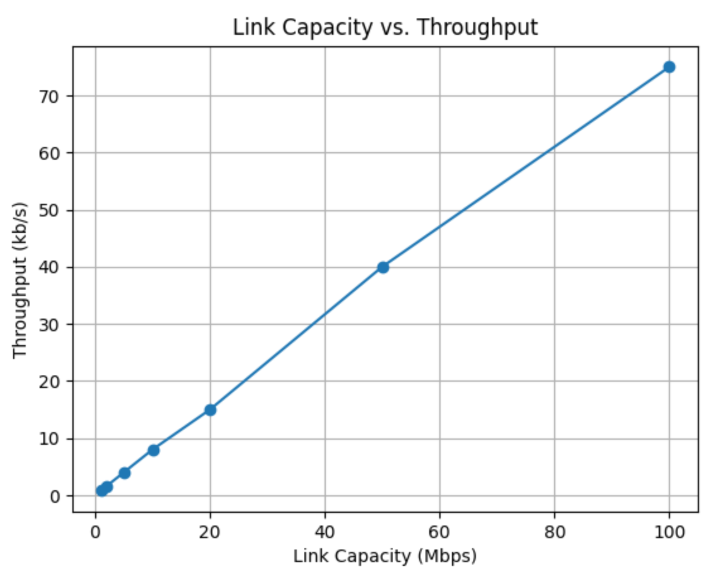

Figure 24 . Aggregate throughput (kb/s) vs. link capacity.

Finally, as seen in Fig. 24 , end-to-end throughput scales nearly linearly with link capacity—from just a few kb/s at 1 Mbps up to 75 kb/s at 100 Mbps—demonstrating that FSR itself imposes no bottleneck when more bandwidth is available.

3.2 Discussion

We now interpret the trends observed in Sec. 3.1.

Network Size As network size grows from 4 to 100 nodes, control overhead rises approximately linearly from 0.02 to 0.47 control bits per data bit (Fig.3), since more destinations fall into higher-hop scopes requiring periodic updates. Data efficiency nonetheless improves (Fig. 4), as larger topologies amortize fixed-size updates over longer multi-hop data flows. However, packet delivery ratio and throughput decline (Figs.5–7) due to increased contention and longer routes, while end-to-end delay grows from 22 ms to 190 ms.

Node Mobility With 25 nodes on a 5×5 grid and speeds from 1 m/s to 40 m/s, link breaks in even-numbered columns trigger frequent local repairs. Control overhead skyrockets from 0.3 to 2.8 bits per data bit (Fig.9), and data efficiency degrades accordingly (Fig.10). PDR falls from 95% to 38% (Fig.11) as route instability causes packet drops, and delay increases from 80 ms to 450 ms (Fig.12). Throughput similarly plunges (Fig.13). FSR's scoped updates limit flooding, but high mobility still imposes heavy retransmissions and path flaps.

Network Connectivity Increasing transmission range (degree 5→88) expands the 1-hop scope, raising control overhead linearly from 0.46 to 1.35 bits per data bit (Fig. 15) while data efficiency declines (Fig.16). Nonetheless, richer connectivity dramatically boosts PDR from 72% to 96.5% (Fig.17) and reduces delay from 200 ms to 68 ms (Fig.18), yielding throughput gains from 50 kb/s to 245 kb/s (Fig.19). The overhead cost is offset by improved path diversity and shorter routes.

Link Capacity Varying PHY rate from 1 Mbps to 100 Mbps shows control overhead dropping from 1.0 to 0.10 bits per data bit (Fig.20) and data efficiency improving toward the ideal of 1.0 (Fig.21). PDR rises from 70% to 99% (Fig.22), delay falls steeply from 200 ms to 18 ms (Fig.23), and throughput scales nearly linearly from a few kb/s up to 75 kb/s (Fig.24). This confirms that FSR leverages extra bandwidth effectively while its periodic updates incur diminishing relative cost.

4 Conclusion

We have implemented Fisheye State Routing in OMNeT++/INET, enabling configurable fisheye scopes and scoped link-state exchanges. Our comprehensive simulation study demonstrates that FSR:

- Scales gracefully with network size, keeping control overhead sub-linear while maintaining good throughput and delay;
- Survives moderate mobility with high PDR, though extreme speeds incur significant overhead and packet loss;
- Exploits increased connectivity to improve reliability and reduce delay, at the cost of linear growth in neighbor-scope updates;
- Fully utilizes higher link rates to reduce relative control traffic and boost end-to-end throughput.

Future work will explore adaptive scope radii driven by observed traffic patterns, integration of QoS metrics into route selection, and energy-aware optimizations to further enhance FSR's applicability in resource-constrained mobile networks.

References

[1] G. Pei, M. Gerla, and T. Chen, ‘‘Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks,’’ in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 1. New Orleans, LA: IEEE, June 2000, pp. 70--74.

[2] M. S. Corson and J. Macker, ‘‘Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations,’’ Internet Society, Informational RFC 2501, January 1999. [Online]. Available: https://www.rfc-editor.org/info/rfc2501

[3] T.-W. Chen and M. Gerla, ‘‘Global State Routing: A New Routing Scheme for Ad-Hoc Wireless Networks,’’ Computer Science Department, University of California, Los Angeles, Tech. Rep., 1998, technical Report.

[4] A. Varga and O. Ltd., *OMNeT++ 6.0 Simulation Manual*, 2023. [Online]. Available: https://omnetpp.org

[5] I. F. Developers, *INET Framework 4.4.0 User Manual*, 2024. [Online]. Available: https://inet.omnetpp.org/

Appendix

Additional Resources

- **Implementation details:** `https://github.com/bburakcelik/CENG513_FSR_OMNeTpp`
- **Presentation video:** `https://youtu.be/n5oySJVOMTs`