

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



Fisheye State Routing Simulation in OMNeT++

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

Prepared by
Mohamed Eldeeb
Student ID: 2643120
mohamed.eldeeb@metu.edu.tr
Computer Engineering
15 June 2025

Abstract

This report details the implementation and performance evaluation of the Fisheye State Routing (FSR) protocol within the OMNeT++ simulation environment, utilizing the INET Framework. FSR, a proactive routing protocol for mobile ad-hoc networks, aims to reduce routing overhead by employing a fisheye approach to information propagation. The report investigates FSR's behavior under diverse network conditions by varying parameters such as link capacity, traffic load, network size, and node mobility speed. Key performance metrics, including end-to-end throughput, delay, packet delivery ratio, and routing overhead, were analyzed. The results indicate that FSR's effectiveness is highly dependent on specific network conditions and its own scope update frequencies. Generally, lower scope refresh rates tended to yield better performance by reducing control packet overhead. Network size and the rate of topological change due to mobility were also found to be critical factors influencing FSR's ability to maintain stable and efficient routes. Notably, a comparison with Global State Routing (GSR) revealed that the fisheye approach did not consistently provide significant performance benefits, suggesting that reduced control overhead from FSR does not always translate to better overall performance in the tested scenarios.

Table of Contents

Abstract	ii
List of Figures	iv
1 Introduction	1
2 Background and Related Work	1
2.1 OMNeT++ Network Simulator	1
2.2 Fisheye State Routing	1
3 Implementation	1
4 Results and Discussion	2
4.1 Methodology	2
4.2 Results	3
4.2.1 Link capacity	3
4.2.2 Node count	3
4.2.3 Message length	5
4.2.4 Message send interval	6
4.2.5 Mobility speed	7
5 Conclusion	8

List of Figures

Figure 1	Simulation environment in OMNeT++	2
Figure 2	Link capacity (bitrate)	4
Figure 3	Node count	5
Figure 4	Message length	6
Figure 5	Mean of send interval (exponential distribution)	7
Figure 6	Minimum speed of mobility	8

1 Introduction

Fisheye State Routing (FSR) is a proactive routing protocol designed for mobile ad-hoc networks. It uses a fisheye approach to reduce the amount of routing information exchanged between nodes, allowing for efficient routing in dynamic topologies. This report presents the implementation and evaluation of FSR using the INET Framework in OMNeT++. The simulation results demonstrate the performance of FSR under various network conditions, including different link capacities, traffic loads, network sizes, and topology changes.

2 Background and Related Work

2.1 OMNeT++ Network Simulator

OMNeT++ [1] is a discrete event simulator that provides an environment for simulating networks. The INET Framework addon includes implementations of various Internet and wireless technologies, including the IP stack and IEEE 802.11.

2.2 Fisheye State Routing

Fisheye State Routing (FSR) is a proactive routing protocol designed for mobile ad-hoc networks, proposed by Pei et al. in [2]. It is based on link state routing, where the nodes maintain a state of the network topology and exchange routing information periodically. FSR uses a fisheye approach to reduce the amount of routing information exchanged between nodes, allowing for efficient routing in dynamic topologies.

Nodes exchange their link states with their neighbors periodically, but each node only sends updates for links that are within a certain scope. The scope is defined by the distance (in hops) to the node, with closer nodes receiving more frequent updates. This allows for a more efficient use of bandwidth and reduces the overhead in the control plane.

3 Implementation

The INET framework provides a modular architecture for implementing network protocols. The FSR protocol is implemented as a routing module that can be configured and used in simulations. The implementation stack was based on the structure used in existing routing modules such as OSPF.

The FSR module is implemented in C++ and consists of several components, including the routing table, the fisheye scope management, and the message handling logic. The routing table maintains the state of the network topology, using Dijkstra's algorithm for shortest path calculation, while the fisheye scope management determines which links to include in the link state update packets (LSUs) based on their distance from the node. The message handling logic processes incoming LSUs and updates the routing table accordingly.

The implementation is open source under the GPLv3 license and can be found on GitHub at <https://github.com/mosamadeeb/inet-fsr> [3].

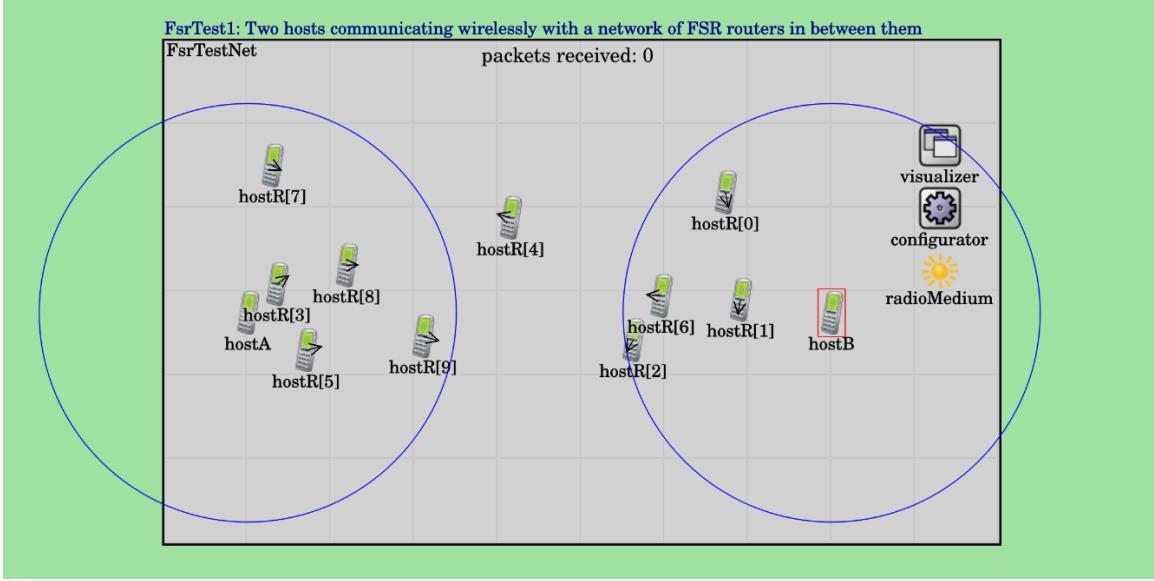


Figure 1 . Simulation environment in OMNeT++

4 Results and Discussion

4.1 Methodology

The setup used for evaluation is based on the Wireless Tutorial for INET [4], using CSMA/CA for medium access control, although the path loss and interference modules were disabled as the focus is on the routing performance.

Figure 1 shows the simulation setup in the OMNeT++ IDE. The simulation environment consists of a single host (A) sending messages to another host (B) through one or more relay nodes (R_1, R_2, \dots) in between. Hosts A and B are stationary, placed 700m apart, while the other nodes are free to move in an 800mx400m rectangular area centered in the middle between the two main hosts. The maximum communication range of each node is 250m, so on average two hops will be needed for host A to reach host B. The mobile nodes follow the random waypoint mobility model; after reaching a random position within the constrained area, the nodes remain stationary for 3 seconds before moving again. The speed of mobility is a random variable that is uniformly distributed.

The simulation runs for 90 seconds, with the first 2 seconds used for warm-up to allow the routing protocol to stabilize. Each study is run with 30 replications to ensure statistical significance, and the mean is taken along with 95% confidence intervals.

The parameters varied in the simulation are link capacity, message length, message send interval, relay nodes count, and mobility speed. Studies vary a single variable while keeping the others constant. In each study, the non-varied parameters use these default values:

- Link capacity: 8 Mbps
- Message length: 1000 bytes
- Message send interval: 150 milliseconds (mean of exponential distribution)
- Relay nodes count: 10
- Mobility speed: 10 to 20 m/s (uniformly distributed)

Each study also tests four different configurations of the FSR protocol, with different periods for the scope updates. All of them have three scopes except the last one, which has only one, to represent Global State Routing (GSR). The scope update periods are as follows:

- 250, 500, 750ms
- 500, 750, 1000ms
- 750, 1500, 2250ms
- 500ms (GSR)

The performance metrics collected include end-to-end throughput, end-to-end delay, packet delivery ratio, data bits transmitted per data bit delivered (i.e. number of hops to reach destination), and control bits transmitted per data bit delivered. These metrics are taken from the simulation results generated by INET. For end-to-end throughput and delay, the values are taken directly from the UDP app module installed in the main hosts. The other metrics are calculated based on the packet sizes recorded in the `*.wlan[0].mac` module in all nodes, which records the packets transmitted to the radio medium.

4.2 Results

4.2.1 Link capacity

Figure 2 shows the effects of link capacity on various metrics.

Higher link capacities lead to higher throughput, although the effect is not that significant for values higher than 2 Mbps, as seen in Figure 2 a. The insignificant but still increasing effect past that point might be related to efficiency in the transmission of control packets.

Link capacity has a direct effect on lowering delay. The effect is diminished as the capacity increases past 4 Mbps however.

As with throughput, the packet delivery ratio increases with higher link capacity up to a certain optimal point, then it levels off.

The ratio of data bits transmitted per bit delivered is higher for lower link capacities, as shown in Figure 2 d. This is expected, as higher link capacities allow for more efficient transmission of data packets, leading to fewer retransmissions before routes change.

For control bits transmitted per data bit delivered, the effect is similar to that of data bits, as shown in Figure 2 e. What is interesting is the differences between the FSR configurations, where those with higher refresh rates resulted in a higher control-to-data ratio. As for the GSR configuration, it did not show much of a difference compared to the corresponding FSR configuration with the same update interval for the first scope (500ms). This might indicate that this simulation scenario relied greatly on first scopes, and that the second and third scopes did not affect the routing performance much.

4.2.2 Node count

Figure 3 shows the effects of node count on the tested metrics.

Throughput decreased with increasing node counts, as shown in Figure 3 a, possibly due to increased contention for the shared medium.

Node count has a negative effect on end-to-end delay. This is more noticeable in instances with more frequent FSR updates, as the least frequent FSR instance (750ms for the first

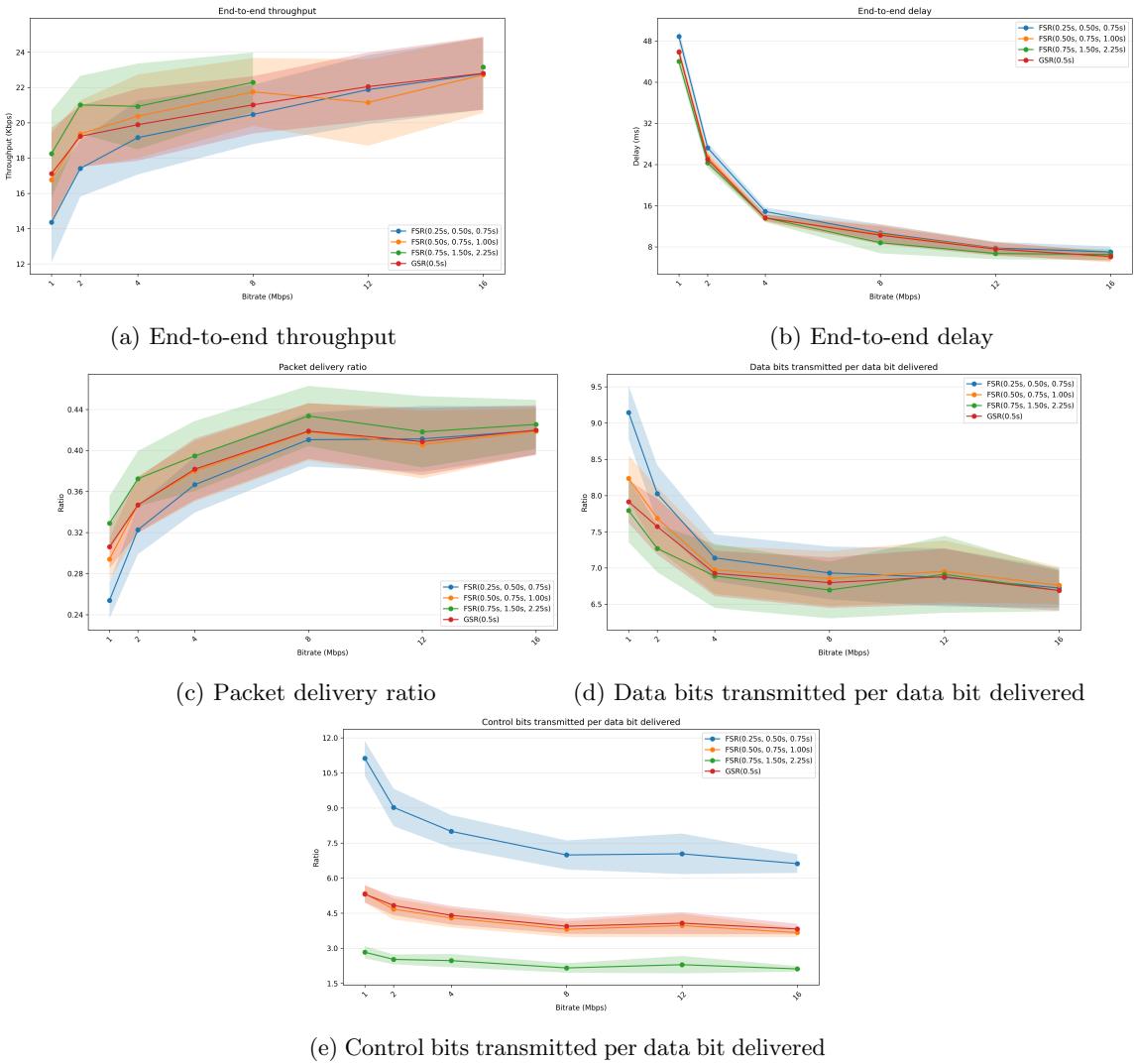


Figure 2 . Link capacity (bitrate)

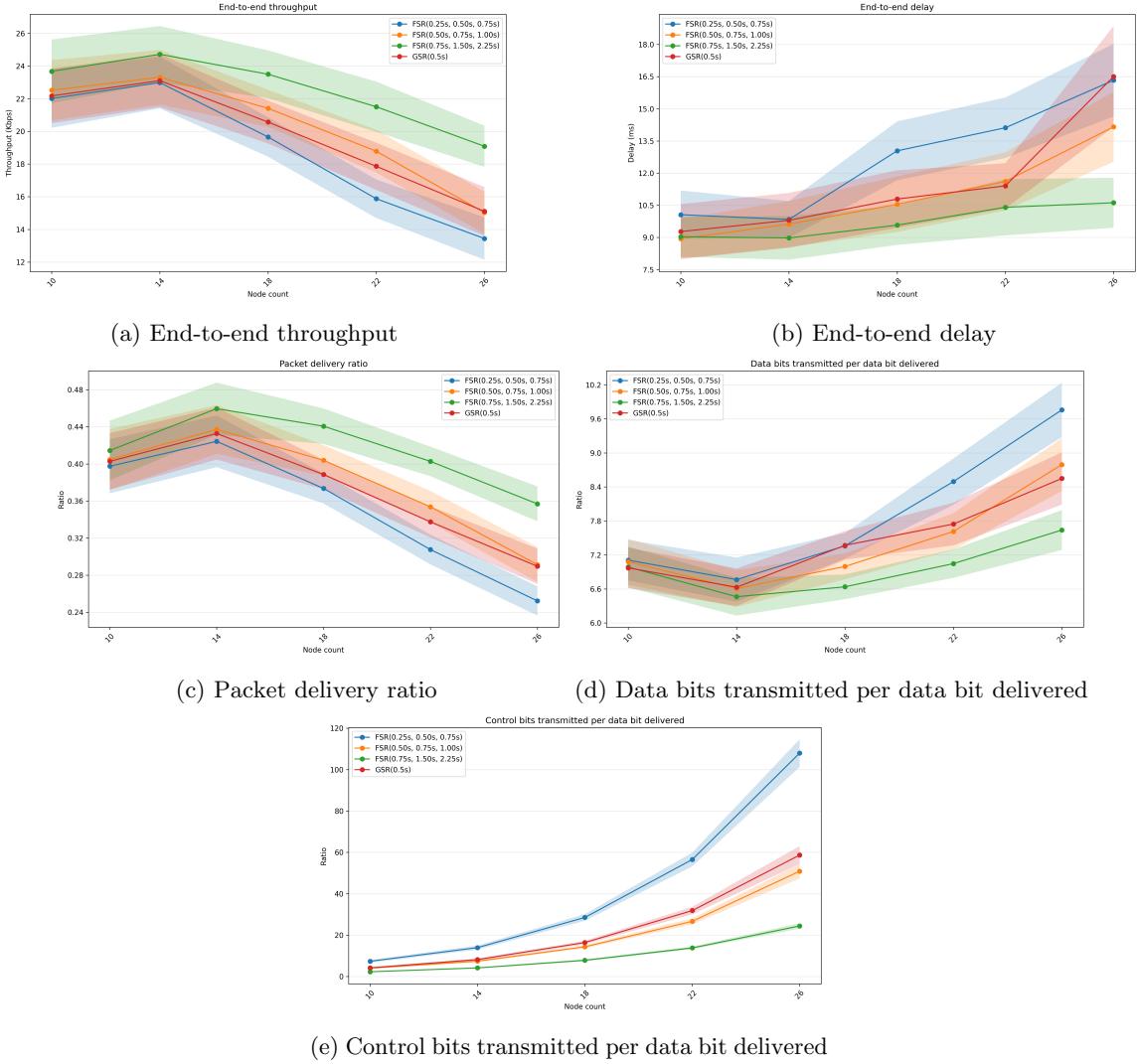


Figure 3 . Node count

scope) does not seem to suffer much from increased node counts. This seems to be a general trend with the other metrics too, where lower refresh rate results in better performance.

For the scenarios tested, it appears that a network size of 14 relay nodes is optimal, as higher node counts lead to lower packet delivery ratios, as shown in Figure 3 c. This is likely due to increased contention for the shared medium, leading to more collisions and retransmissions. Lower node counts also result in worse performance as the number of relay nodes is not sufficient to maintain a stable route between the main hosts.

The data and control bit transmission ratios are larger for higher node counts, as shown in Figures 3 d and 3 e. This is expected, as more nodes lead to more hops in the route, resulting in more data and control bits transmitted. We can also see how much of a difference the scope update frequencies affect these results.

4.2.3 Message length

Figure 4 shows the effects of message length on the tested metrics.

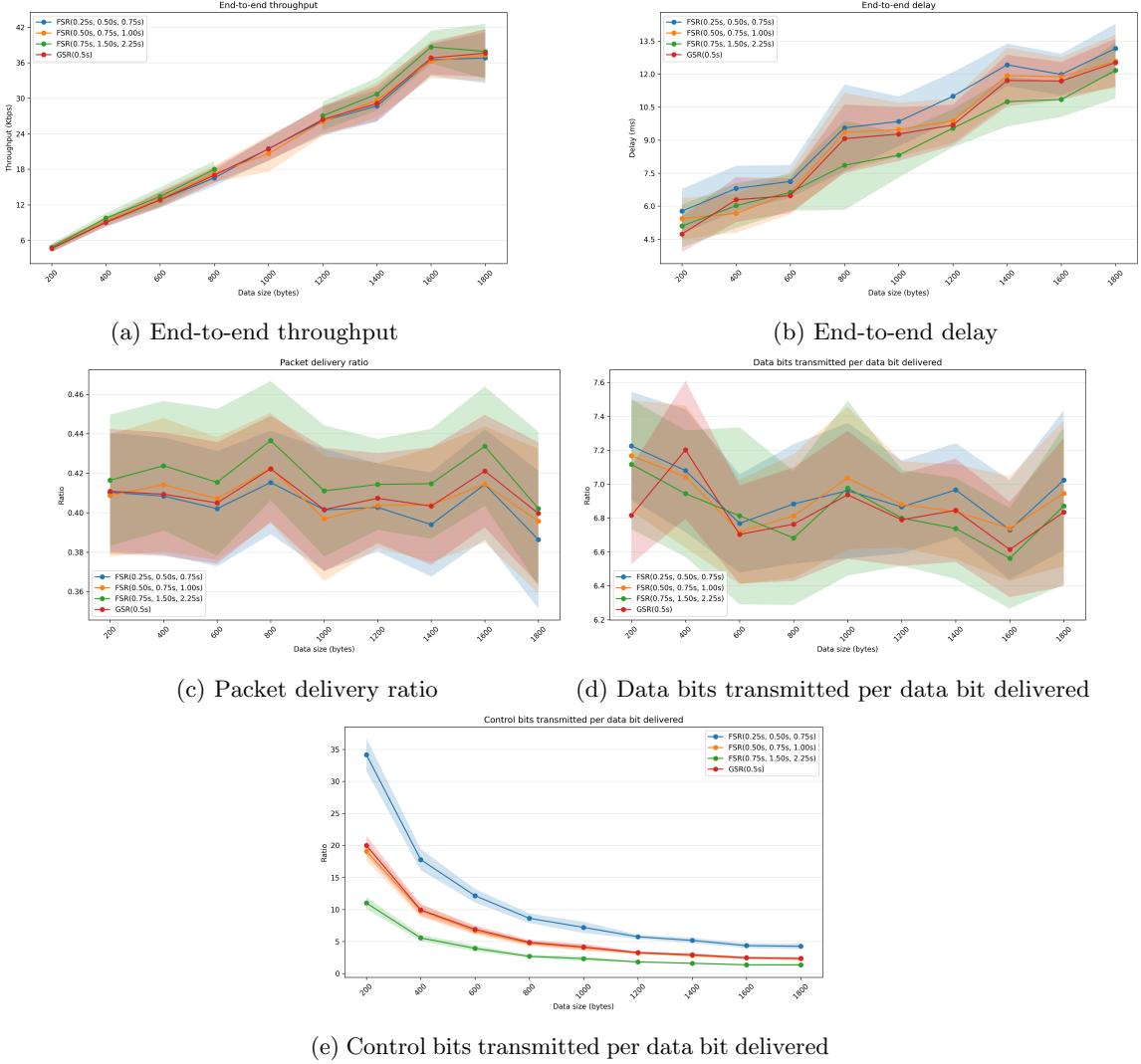


Figure 4 . Message length

Message length has a linear relationship with throughput, as shown in Figure 4 a. The longer the message, the more bits are transmitted, leading to higher throughput. This shows that the default link capacity of 8 Mbps is sufficient for the message lengths used in the simulation.

The effect of message length on end-to-end delay is similar to that on throughput, where larger messages lead to larger delays.

There does not seem to be a consistent trend in the packet delivery and data transmission to delivery ratios, as shown in Figures 4 c and 4 d.

Low message lengths result in a higher ratio of control bits transmitted per data bit delivered, and this is expected as message length directly affects the number of data bits delivered.

4.2.4 Message send interval

Figure 5 shows the effects of message sending interval on the tested metrics.

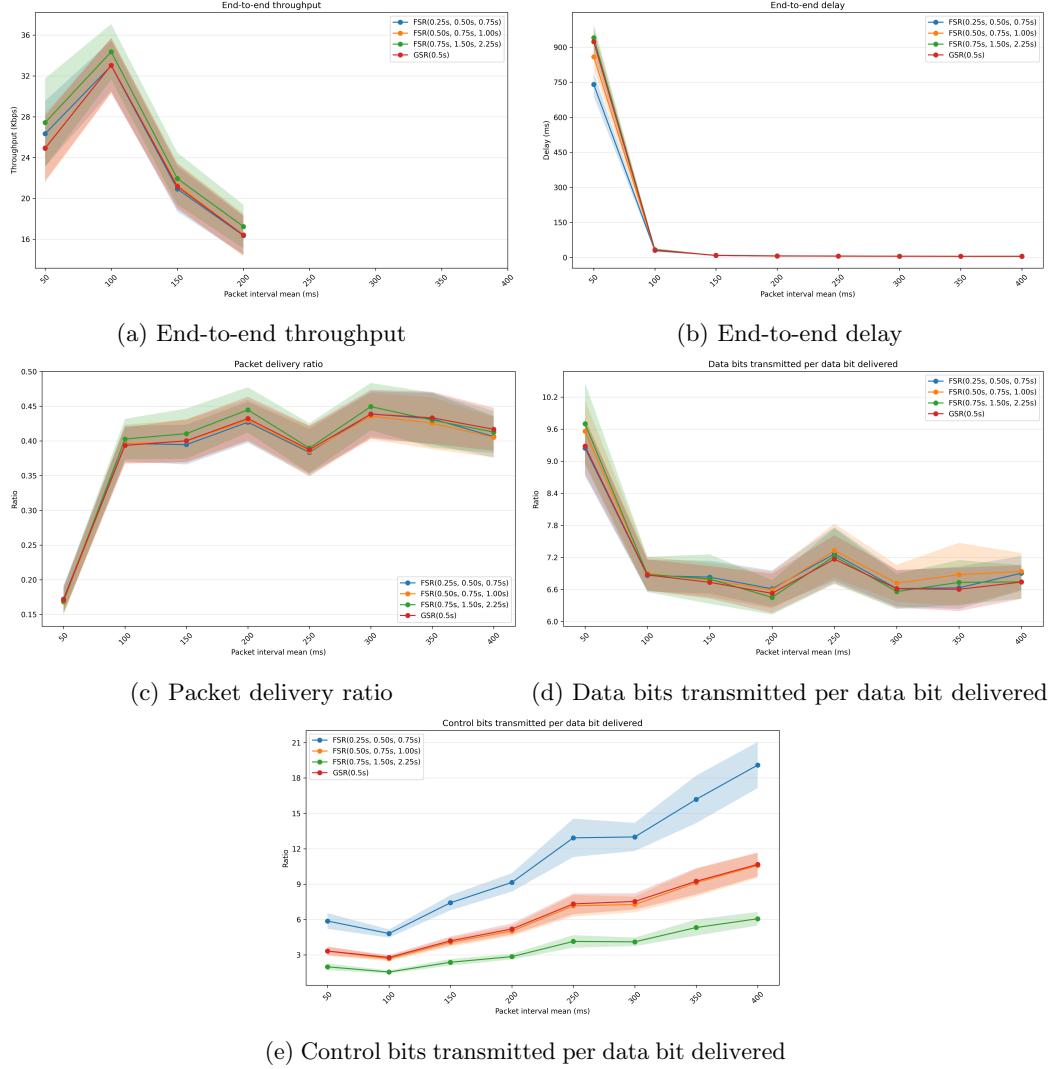


Figure 5 . Mean of send interval (exponential distribution)

Figure 5 a shows the results for message sending interval. Higher intervals lead to lower throughput, although the throughput also decreased at the minimum interval tested (50ms), which is likely due to the increased contention for the shared medium, as well as messages being sent too frequently for the mobile nodes to handle efficiently.

End-to-end delay has an interesting outlier at the lowest value tested, as seen in Figure 5 b. It is likely that the higher transmission demands caused extreme traffic congestion beyond the capacity of the network, which hindered the delivery of messages. This is also visible in the packet delivery ratio and data bit transmission-to-delivery ratios in Figures 5 c and 5 d.

Figure 5 e shows that higher data message sending intervals lead to higher control-to-data bit ratios, as higher intervals reduce the number of data bits transmitted.

4.2.5 Mobility speed

Figure 6 shows the effects of mobility speed on the tested metrics. The mobility values shown are the minimum speed of mobility, and the maximum speed is set to two times the

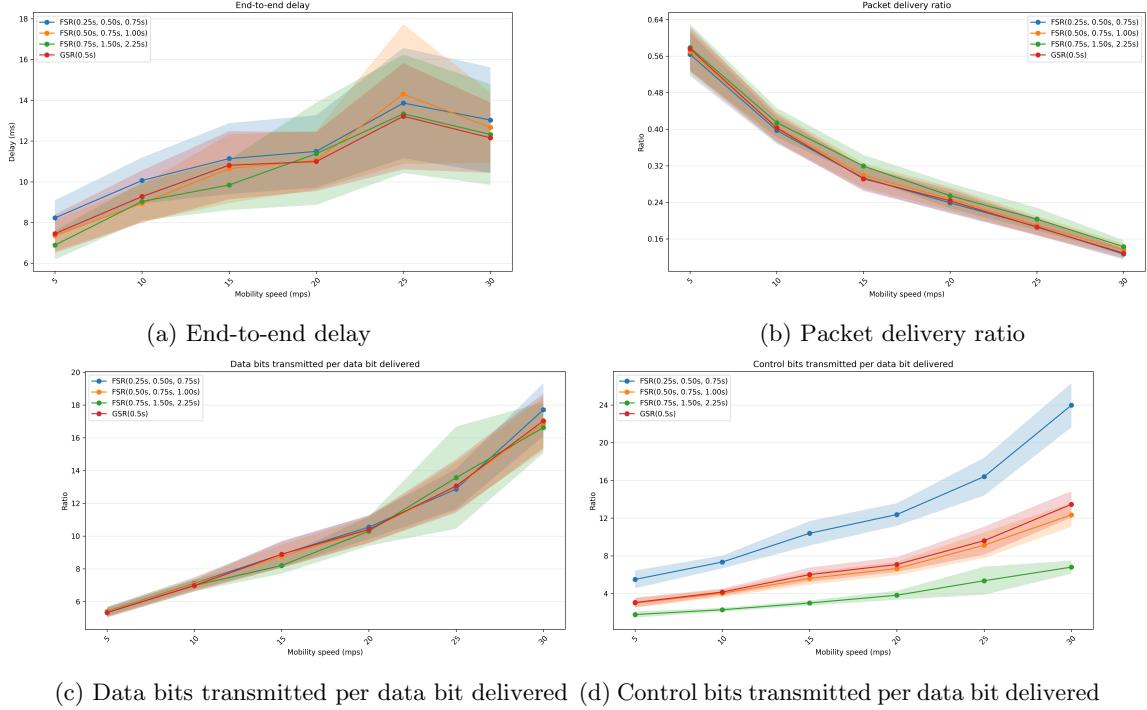


Figure 6 . Minimum speed of mobility

minimum speed.

The effect of mobility speed on end-to-end delay is shown in Figure 6 a. Higher mobility leads to more frequent topology changes, which leads to more changes in routes and higher delays. This is also supported by the large confidence intervals, showing the effect of random mobility in the simulation replications.

Figure 6 b shows that the packet delivery ratio decreases with higher mobility speeds. This is due to the increased likelihood of nodes moving out of range of each other, leading to more route changes and fewer successful transmissions. What is interesting here is that even with higher mobility, the FSR configuration with the lowest scope refresh rate still performed better than the others on average.

The data and control bit transmission ratios are larger for higher mobility speeds, as shown in Figures 6 c and 6 d. As expected, higher mobility results in more topological and routing changes, resulting in less data packets delivered per transmission. For the control-to-data ratio, the effect of mobility speed is less pronounced for configurations with lower scope refresh rates.

It was difficult to measure throughput for different mobility speeds due to some instabilities in the throughput calculation in the UDP app module. For that reason those results are omitted.

5 Conclusion

From the various parameters studied, we can conclude that the effectiveness of FSR is highly dependent on the network conditions and the configuration of the protocol itself. Lower scope

refresh rates lead to better performance in most cases, as they reduce the overhead of control packets and allow for more stable routes. The network size and topological rate of change also play a significant role in the performance of FSR, with larger networks and higher mobility speeds leading to more challenges in maintaining stable routes. Furthermore, the comparison between FSR and GSR shows that the fisheye approach did not yield significant performance improvements, which implies that the expected lower control overhead does not necessarily translate to better performance in all scenarios.

References

- [1] *Omnet++ discrete event simulator*. [Online]. Available: <https://omnetpp.org/>.
- [2] G. Pei, M. Gerla, and T.-W. Chen, “Fisheye state routing: A routing scheme for ad hoc wireless networks,” in *2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record*, vol. 1, Jun. 2000, 70–74 vol.1. DOI: 10.1109/ICC.2000.853066.
- [3] M. Eldeeb, *Inet-fsr*. [Online]. Available: <https://github.com/mosamadeeb/inet-fsr>.
- [4] *Wireless Tutorial — INET documentation*. [Online]. Available: <https://inet.omnetpp.org/docs/tutorials/wireless/doc/>.