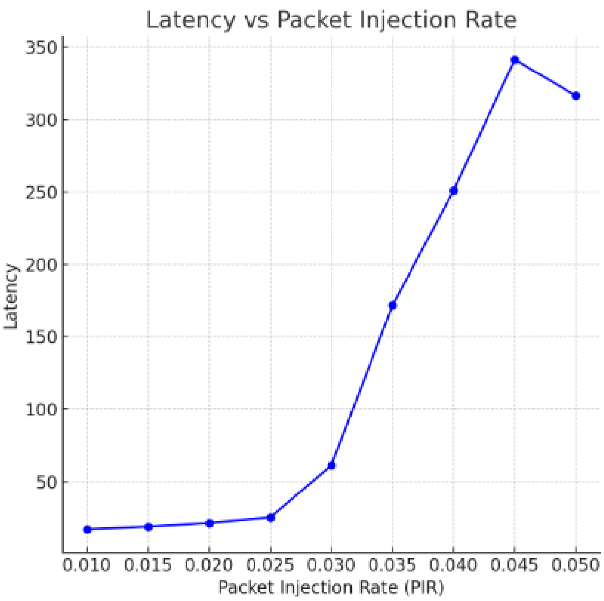
Simulation of Packet Injection Rate and its Effect on Latency and Throughput

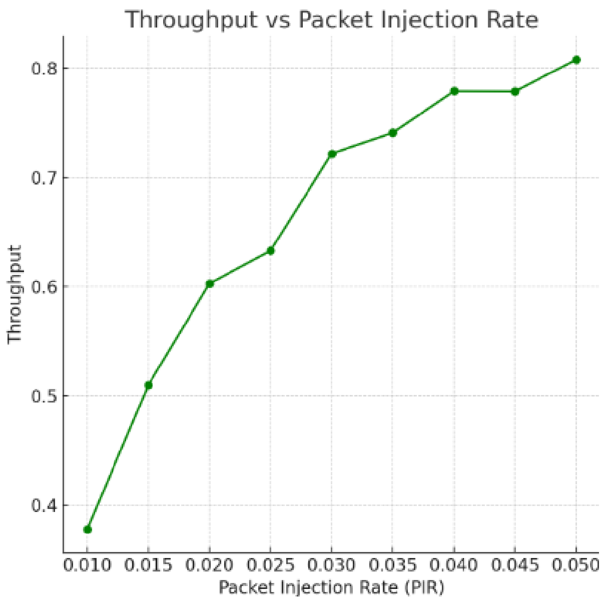
Packet Injection Rate (PIR) varied from **0.01 to 0.05:**

PIR	Latency	Throughput
0.01	17.37	0.378
0.015	19.17	0.51
0.02	21.62	0.603
0.025	25.43	0.633
0.03	61.34	0.722
0.035	171.71	0.741
0.04	251.02	0.7791
0.045	341.33	0.7789
0.05	316.42	0.808

From the above table, performance analysis is done:

The graphs below show the overall change in Latency and Throughput with Packet Injection Rate:





The following can be summarized from the graph and data:

Latency remains low and increases gradually from PIR = 0.01 to 0.025, indicating a stable network performance.

A sharp increase in latency is observed beyond PIR = 0.03, reaching a peak of 341.33 at PIR = 0.045. This indicates congestion in the network.

Throughput increases steadily and reaches a near-saturation point at PIR = 0.05 with a maximum value of 0.808.

A point of interest is observed around PIR = 0.04 to 0.045, where throughput slightly plateaus but latency sharply increases, confirming the onset of saturation and potential network instability.

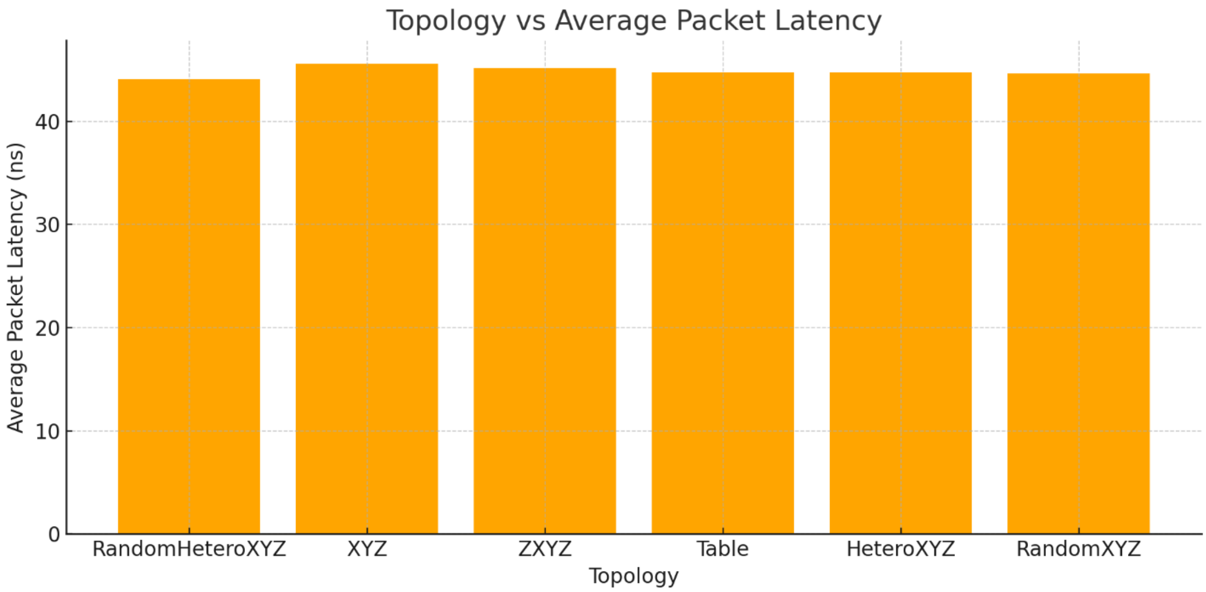
Overall, the results suggest that the network performs efficiently up to a PIR of around 0.03, beyond which the latency significantly increases, signaling a congestion point.

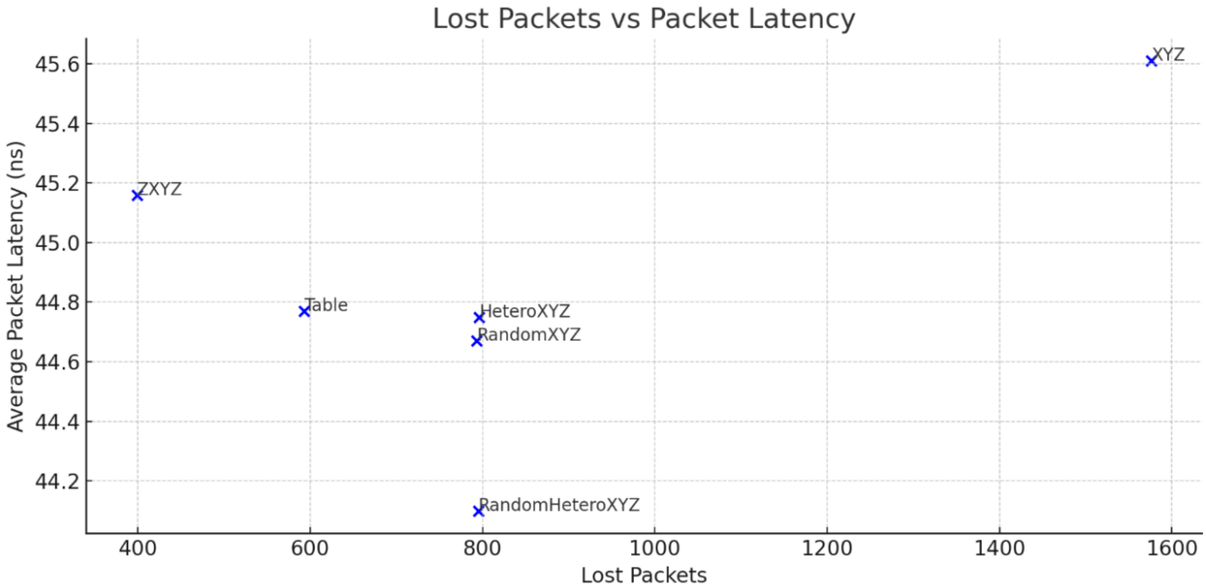
Simulation Results Based on Different Routing Algorithms

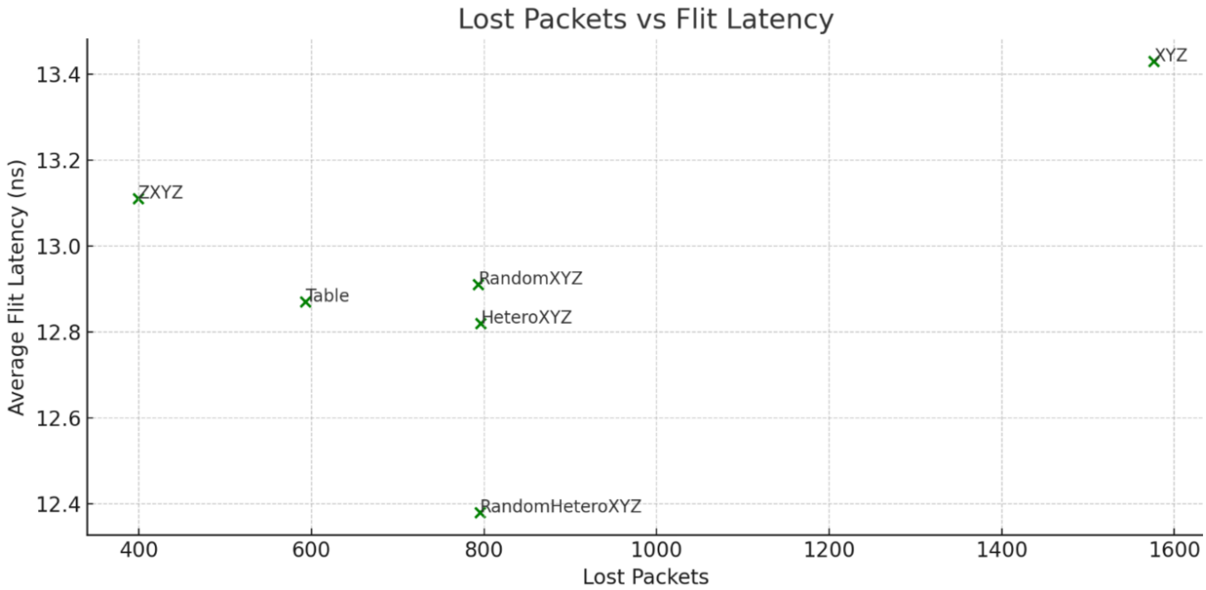
The following data was simulated using Ratatoskr simulator :

Routing Algorithm	Avg. Lost Packets	Avg. Flit Latency (ns)	Avg. Packet Latency (ns)	Avg. Network Latency (ns)	Execution Time Range
RandomHeteroXYZ	792.3	13.02	44.62	79200048.7	7–10 sec
XYZ	913.25	13.29	45.67	91325047.0	9–12 sec
ZXYZ	528.7	13.03	44.74	52866644.0	4–8 sec
Table	791.3	13.19	45.26	79100046.7	9–11 sec
HeteroXYZ	794.5	13.09	45.28	79400048.0	8–10 sec
RandomXYZ	792.5	13.02	45.11	79200048.0	9–10 sec

The graphs below show comparison between important parameters simulated by using different Routing Algorithms:







Based on the data and the graph :

Among all the algorithms:

- **ZXYZ** performed best in terms of **lowest packet loss**
- **RandomHeteroXYZ** and **HeteroXYZ** achieved **efficient latencies** and **fast execution**
- **XYZ** routing exhibited the **highest packet loss**, making it less ideal under load

The selection of the routing algorithm thus depends on the **specific use case**:

- Use **ZXYZ** when **reliability** is top priority
- Choose **RandomHeteroXYZ** or **HeteroXYZ** for **balanced throughput and latency**
- Avoid **XYZ** in scenarios prone to congestion