CHAPTER 4

SNAPSHOTS

4.1 No Interference:

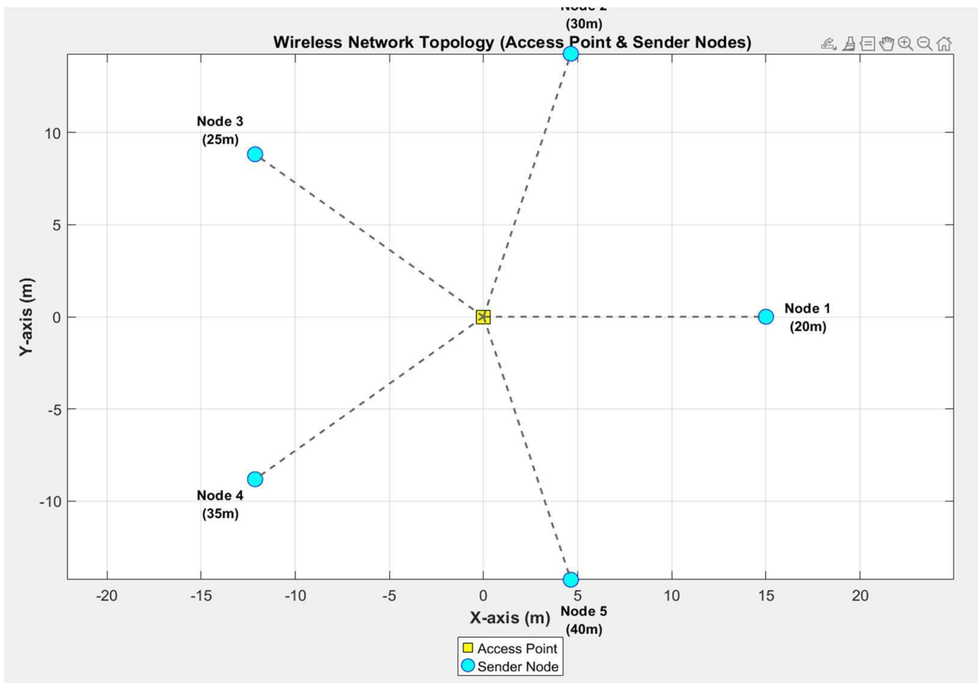

Fig. 4.1.1 No Interference Topology

Wireless Performance Metrics:

TCP Variant	Throughput (Mbps)	Avg Delay (ms)	Jitter (ms)
TCP Reno	1.3412	12.2811	2.2589
TCP New Reno	1.0759	14.9325	1.2306
TCP Tahoe	0.3052	19.3655	0.0881
TCP CUBIC	1.7169	17.6376	0.2255
TCP BIC	0.8590	18.9481	2.2108

Fig. 4.1.2 No Interference Performance metrics

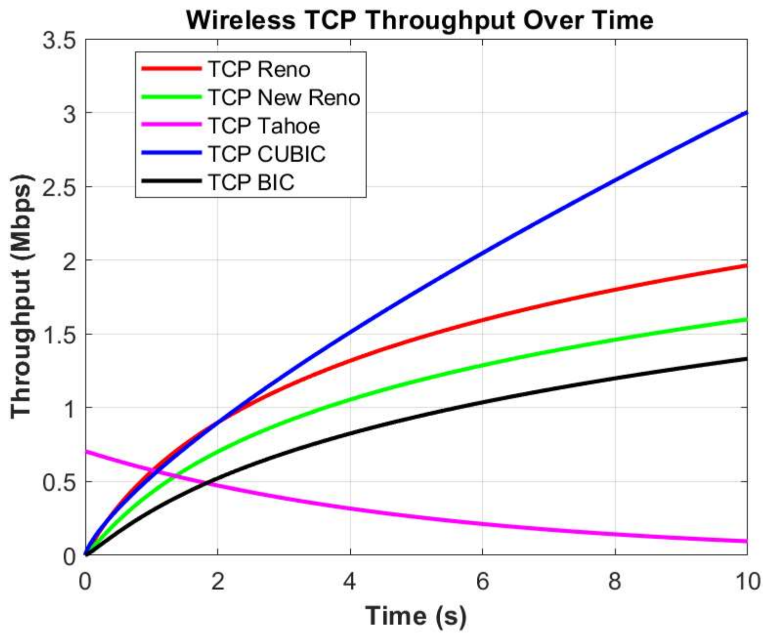

Fig. 4.1.3 No Interference Throughput Graph

4.2 Low Interference:

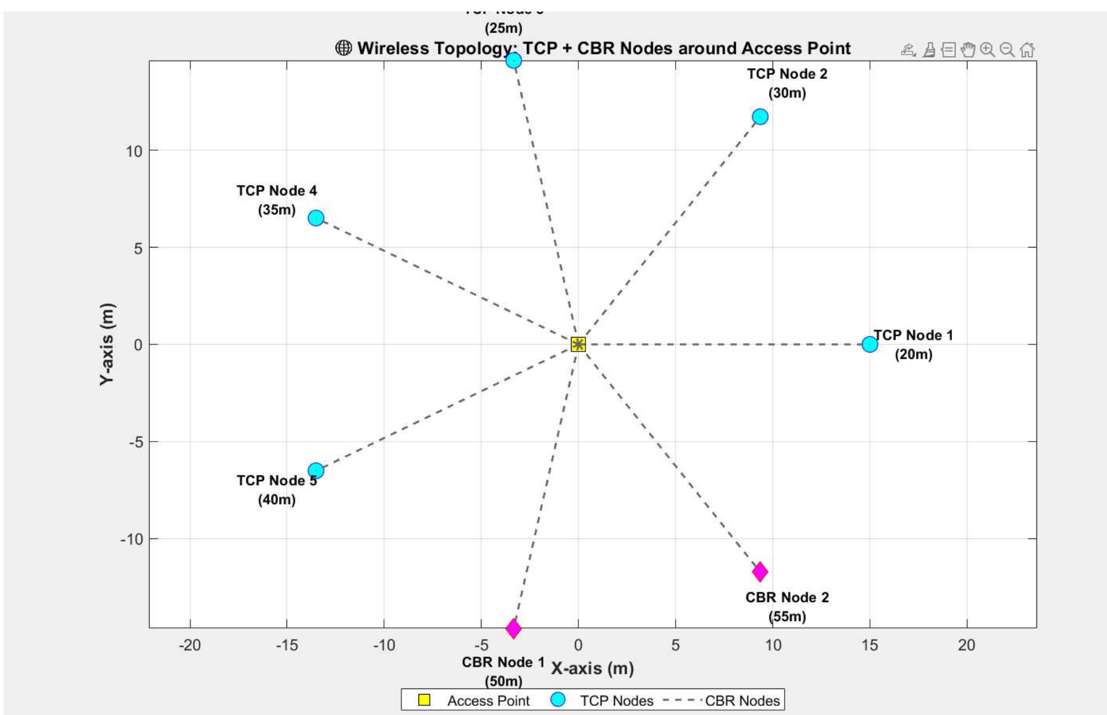

Fig. 4.2.1 Low Interference Topology

Wireless Performance Metrics (TCP + CBR with Interference):

Variant	Throughput (Mbps)	Avg Delay (ms)	Jitter (ms)
TCP Tahoe	0.5732	18.8785	2.5808
TCP BIC	0.7355	25.5861	1.5640
TCP Reno	0.7348	17.6928	0.9325
TCP New Reno	0.5249	20.7982	3.6201
TCP CUBIC	0.6961	24.4968	2.2417
CBR 1	2.5000	26.9838	0.9909
CBR 2	2.3000	27.6140	3.3038

Fig. 4.2.2 Low Interference Performance metrics

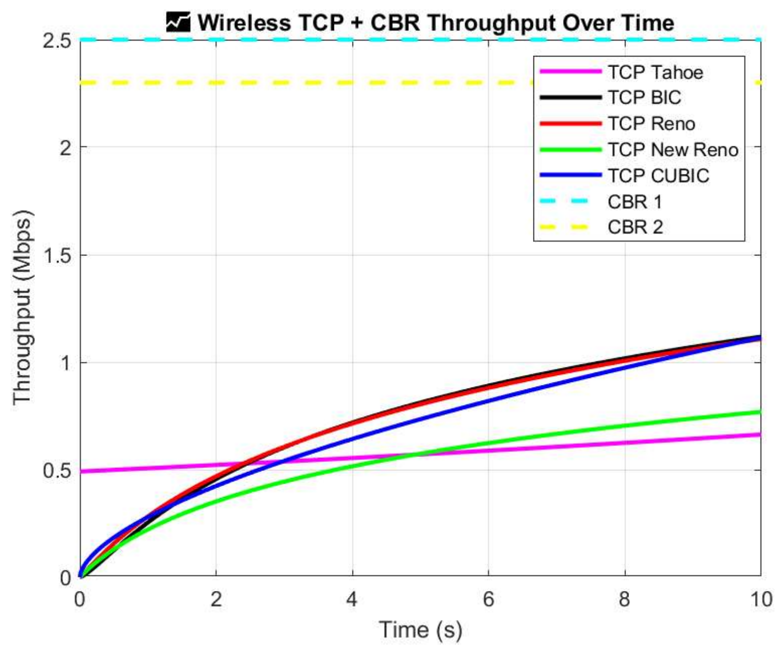

Fig. 4.2.3 Low Interference Throughput Graph

4.3 High Interference:

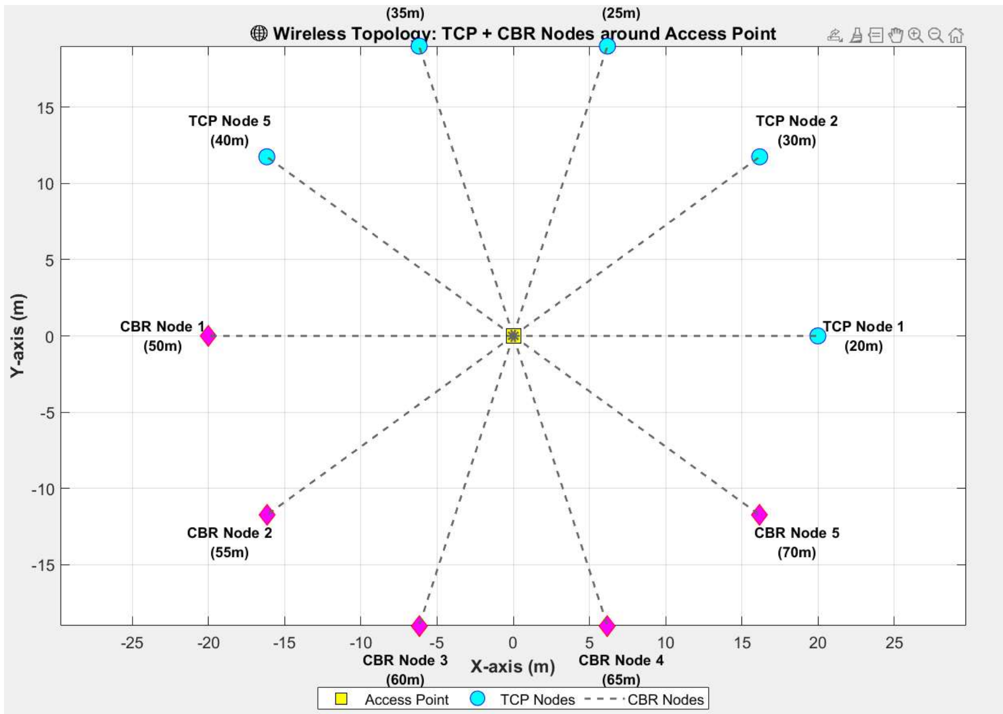

Fig. 4.3.1 High Interference Topology

Wireless Performance Metrics (TCP + 5 CBR Interference, Adjusted Order):

Variant	Throughput (Mbps)	Avg Delay (ms)	Jitter (ms)
TCP BIC	0.3515	29.2172	4.7701
TCP New Reno	0.4142	23.1956	7.8693
TCP CUBIC	0.3115	22.4619	1.0303
TCP Tahoe	0.1097	11.4689	2
TCP Reno	0.3694	19.6564	0.8706
CBR 1	2.5000	29.1054	0.7004
CBR 2	2.3000	32.0370	1.1846
CBR 3	2.1000	38.5914	0.2854
CBR 4	1.9000	45.2039	7.3608
CBR 5	1.7000	45.8017	10.6216

Fig. 4.3.2 High Interference Performance metrics

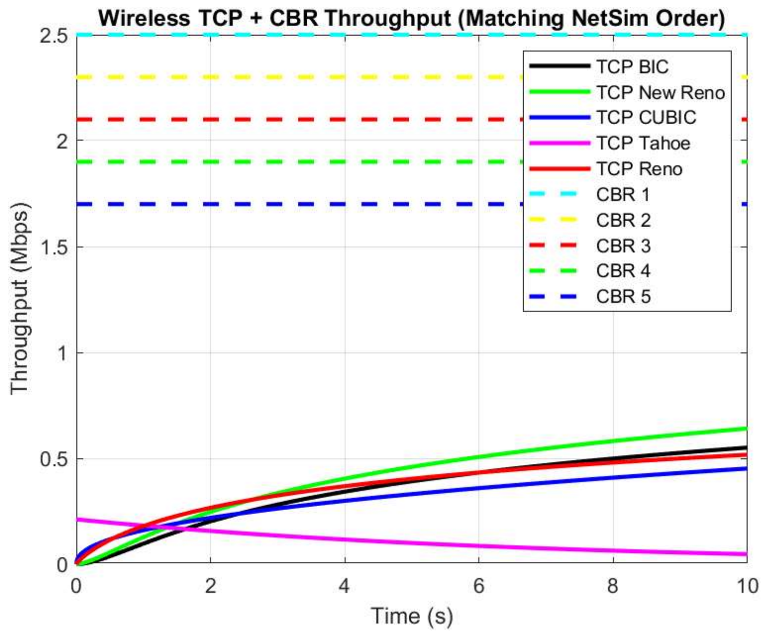

Fig. 4.3.3 High Interference Throughput Graph

4.4 QOS:

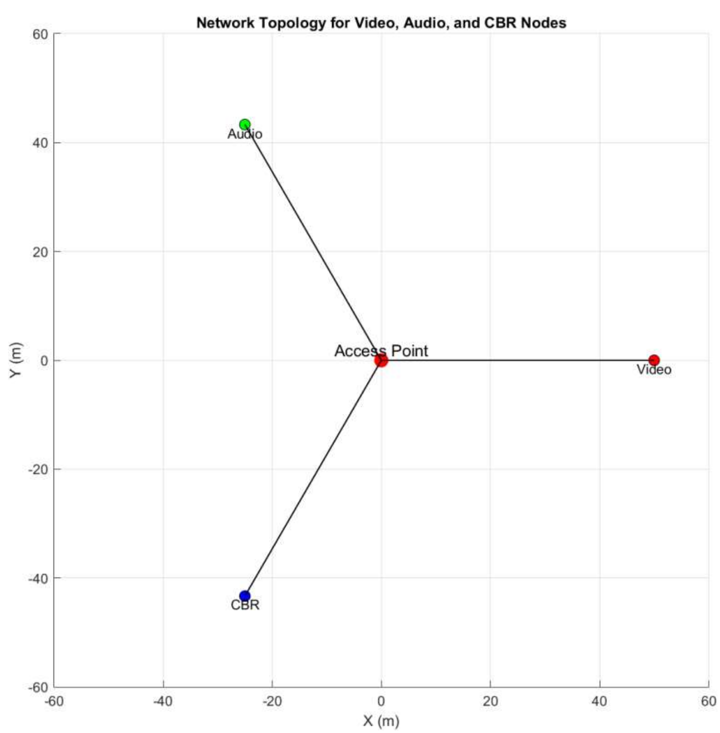

Fig. 4.4.1 Quality of Service Topology

```
--- QoS Summary ---
Video: Avg Delay = 41.88 ms, Avg Jitter = 10.40 ms, Avg Throughput = 2.50 Mbps
Audio: Avg Delay = 208.94 ms, Avg Jitter = 50.38 ms, Avg Throughput = 0.50 Mbps
CBR  : Avg Delay = 100.00 ms, Avg Jitter = 0.00 ms, Avg Throughput = 1.00 Mbps
```

Fig. 4.4.2 Quality of Service Performance metrics

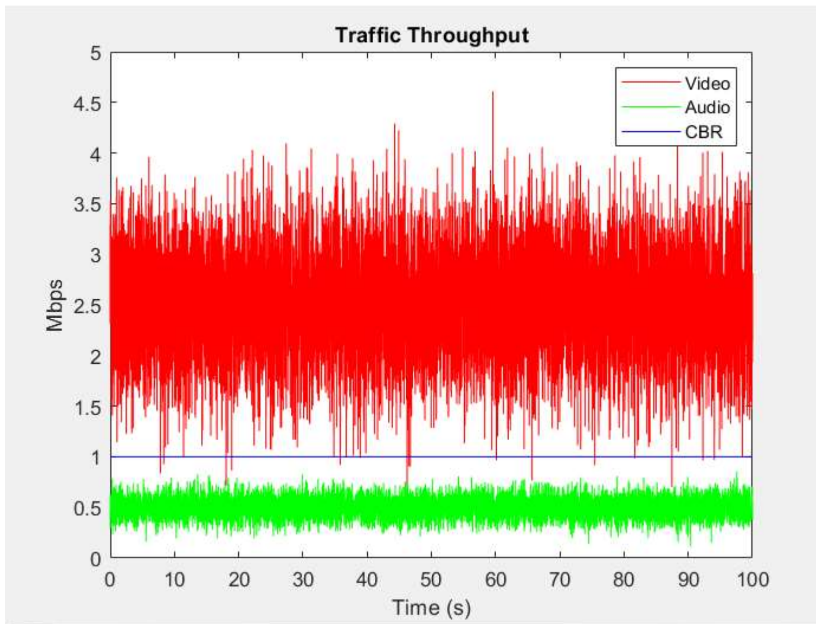

Fig. 4.4.3 Quality of Service Throughput Graph

CHAPTER 5
CONCLUSION AND FUTURE PLANS

Conclusion:

An experiment of TCP implementations across various interference scenarios illustrates varied sensitivity to network dynamics. Under the no-interference scenario, Cubic performs well owing to its aggressive window growth policy, particularly relative to conservative policies like Tahoe, which are penalized by virtue of large backoff intervals. Under low-interference, Reno and BIC perform well, but Cubic and New Reno perform poorly owing to their too aggressive or premature recovery phases. Under high-interference, the protocol performance is severely diminished. Reno and New Reno protocols perform better relative to their counterparts owing to their conservative and predictable reaction to high packet loss. On the other hand, despite Cubic and BIC performing well under the normal case, they perform moderate owing to high retransmissions and poor reaction to persistent interference.

The results validate that the hypothesis no version of TCP can be optimized for all types of performance is true. The pros and cons of each algorithm then now depend on the amount of network interference taking place as well as whether the environment is stable or not.

Hence, the selection of an appropriate congestion control algorithm is critical, as various algorithms react differently to varying network conditions.

This project is to understand the impact of congestion control algorithms and the significance of selecting an algorithm that is tailored for the use-case case scenarios in consideration of the application and the network characteristics of interest. The discussed findings are part of an effort to enhance congestion control for a variety of scenarios; in particular – considering the increasing demands of real-time applications (e.g., video conferencing).

Future plans:

- **Dynamic congestion control using Machine Learning techniques:**
- The future direction of work can include machine learning based congestion control mechanisms which range from simply learning to react to the state of the network. Example include reinforcement learning algorithms that can learn data flow optimization to enhance QoS in scenarios of volatile traffic loads.
- **Simulation on Multiple Platforms like 6 NS-3 OMNeT++:**
- The same experiments can be run and tested on other simulation environments like NS-3 and OMNeT++, to confirm and extend the results. The comparative analysis will in getting a consistent result