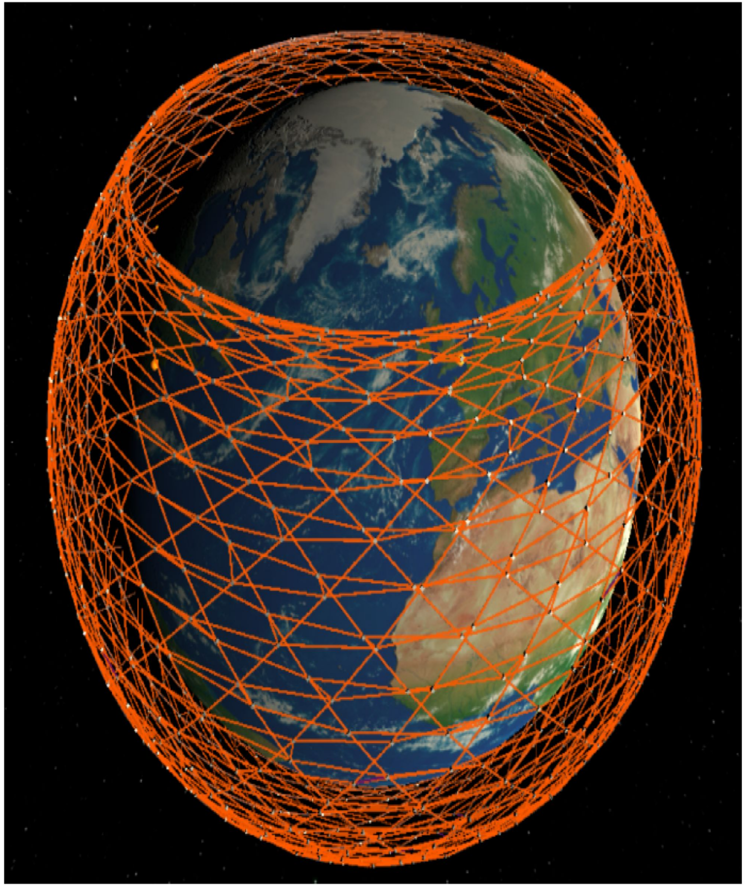
Final Project Presentation

QoS integrated Load-aware routing in LEO Satellites

Presented by
Sumanth Guptha M & Sasank

Mentored by Rahul Agrawal



Agenda

- How LEO satellite works?
- QoS
- Clustering
- Arima
- HBPR
- Results
- conclusion

How does LEO Satellite communication Works?

- User sends the request via user terminal.
- Signals are transmitted and received to the nearest satellite in the range which then relays the signal to a ground station for further transmission.
- Ground station/earth station fetched the data and send it back to the satellite.
- Satellite sends the data back to user terminal, completing the request.
- As satellite moves out of range, the data is handed over to the next satellite that is coming into the range.
- This handover process ensures continuous communication without interruption, even as the user/satellite moves.

Quality of Service

- it refers to the ability of the network to manage and prioritize different types of data traffic—such as real-time communication, streaming media, transactional data, IoT transmissions, and bulk data transfer

→**Real-time Data:** Needs **low latency** and **high reliability** for smooth interaction.

→**Streaming Media:** Requires **high bandwidth** and **low packet loss** for seamless playback.

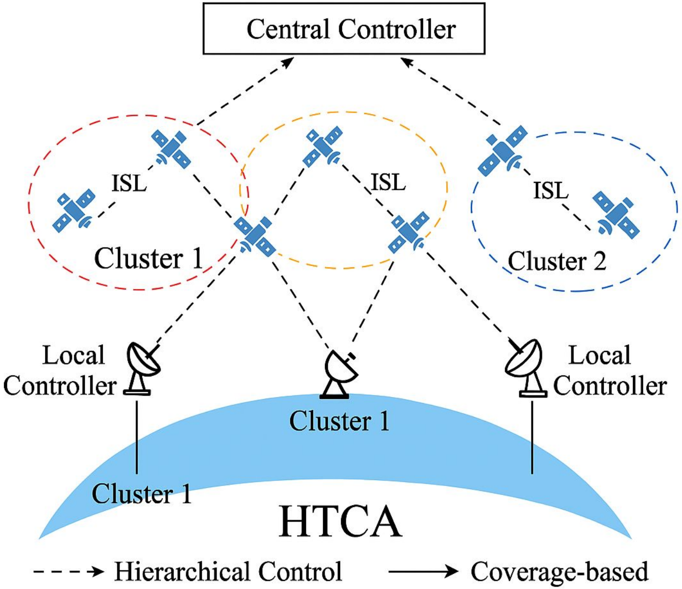
→**Transactional Data:** Demands **low latency** and **high security** for accurate processing.

→ **IoT & M2M:** Prefers **low power use** and **efficient bandwidth** for sensor communication.

→ **Bulk Data Transfer:** Focuses on **high throughput** and **data integrity** for large files.

Clustering

- Each terrestrial ground station functions as a local controller. Because each ground station can only communicate with a limited number of LEO satellites forms a dynamic cluster of satellites within its communication range.
- This set of LEO satellites under a local controller's range is effectively a cluster.
- Local controllers perform network probing using LLDP packets to identify ISLs (inter-satellite links) and collect their subnet's topology.
- We will form 5*4 rectangular clusters centered around ground stations.
- No global routing
- Instead of routing packets across the global through satellites only intra-clustering routing will be done via satellites.



HTCA

Arima(Autoregressive integrated moving average)

- Arima is a statistical analysis model and that uses time series data to either understand the dataset or to predict the future trends.

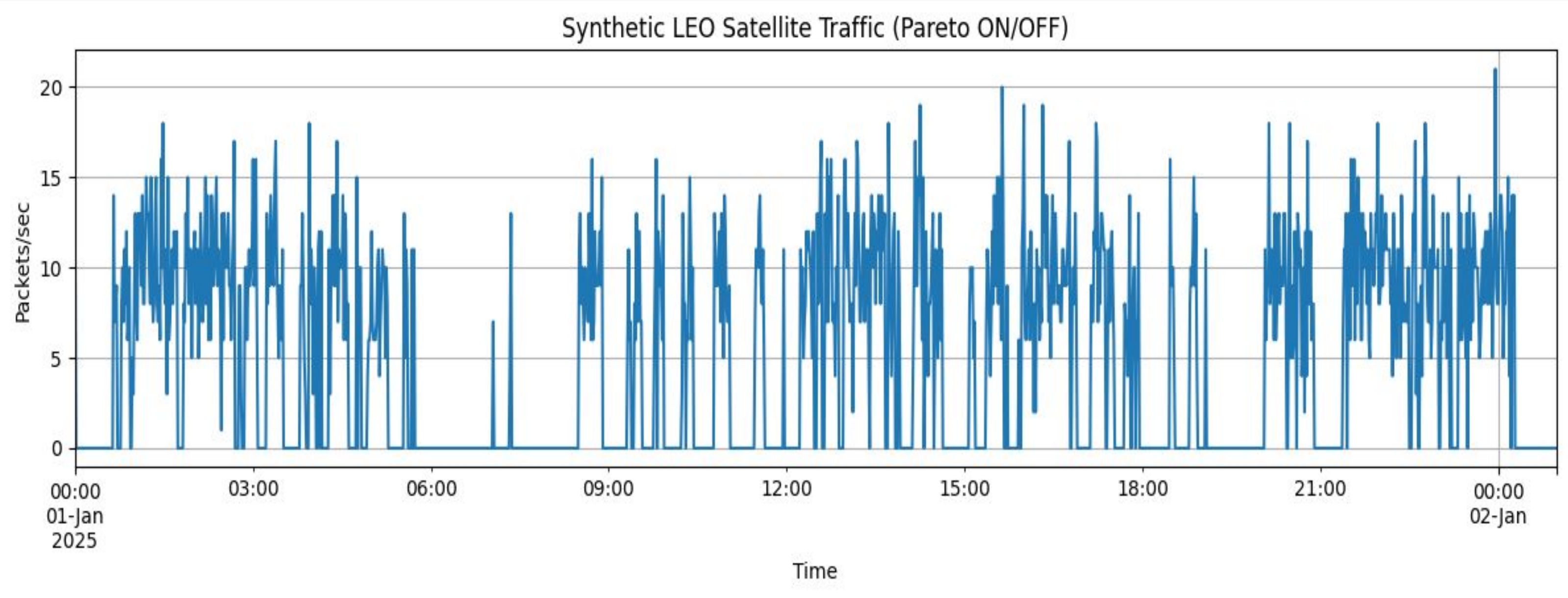
- It has three parameters. Those are
  - P - Autoregressive order
  - D - Seasonal differencing
  - Q - Seasonal Moving Average Order

- For the best results we have used sarima and pmdarima.

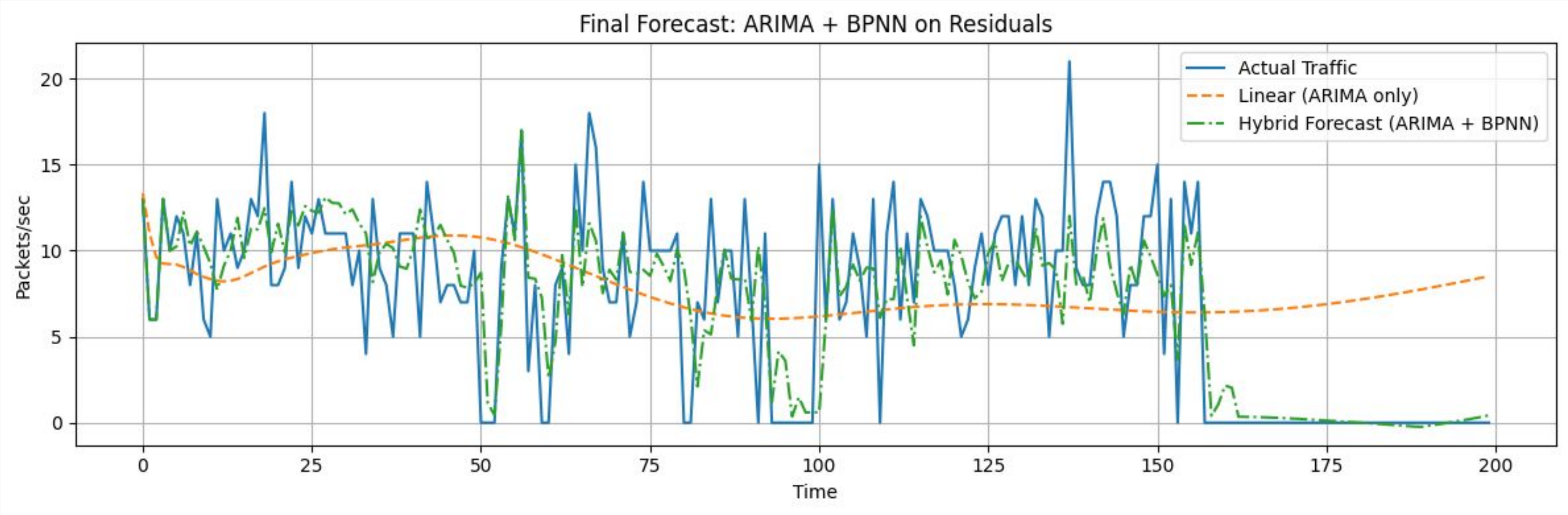
Simulation part of arima

- Imagine you are analyzing network traffic data from a Low Earth Orbit (LEO) satellite. The traffic is bursty, meaning it has periods of high activity (ON) followed by periods of low activity (OFF).
- **How the code works**
  - Traffic Generation
  - Signal Decomposition
  - Linear Forecasting
  - Residual Analysis
  - Non-linear Forecasting
  - Hybrid Forecasting
  - Evaluation

Simulation part of arima



Simulation part of arima



Traditional Queue Length Back-Pressure Routing

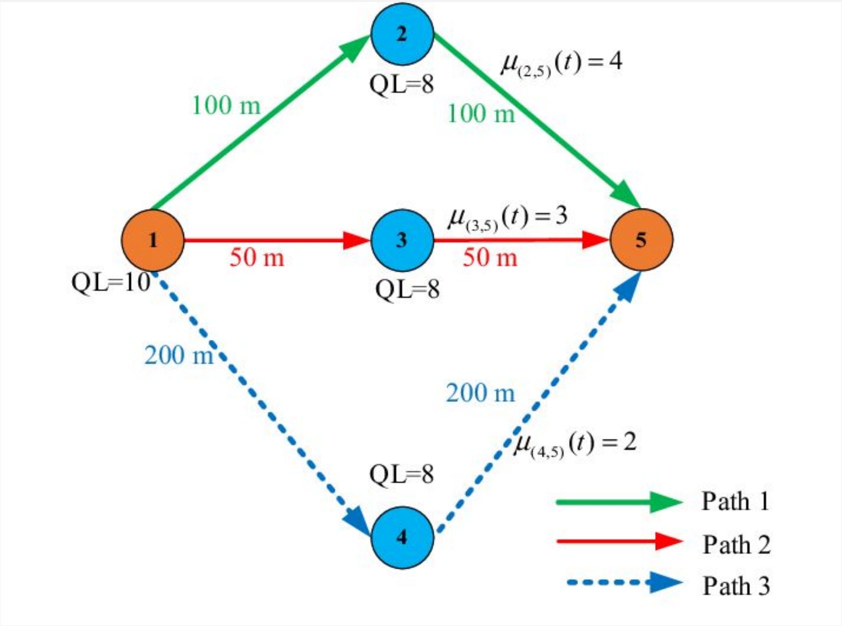
→ Criterion of queue backlog maximization

$$D_{ab}(t) = \max_{c:(a,b)}[P_a^c(t) - P_b^c(t)]$$

$$\text{Maximize} : \sum_{a=1}^{N}\sum_{b=1}^{N} \mu_{ab}(t) D_{ab}(t) \quad \text{s.t.} \mu_{ab}(t) \in \Gamma_{s(t)},$$

$\mu_{ab}(t)$ denotes the transmission rate of link (a, b).

→ BP routing is suitable for all kinds of multi-hop networks to maximize throughput.



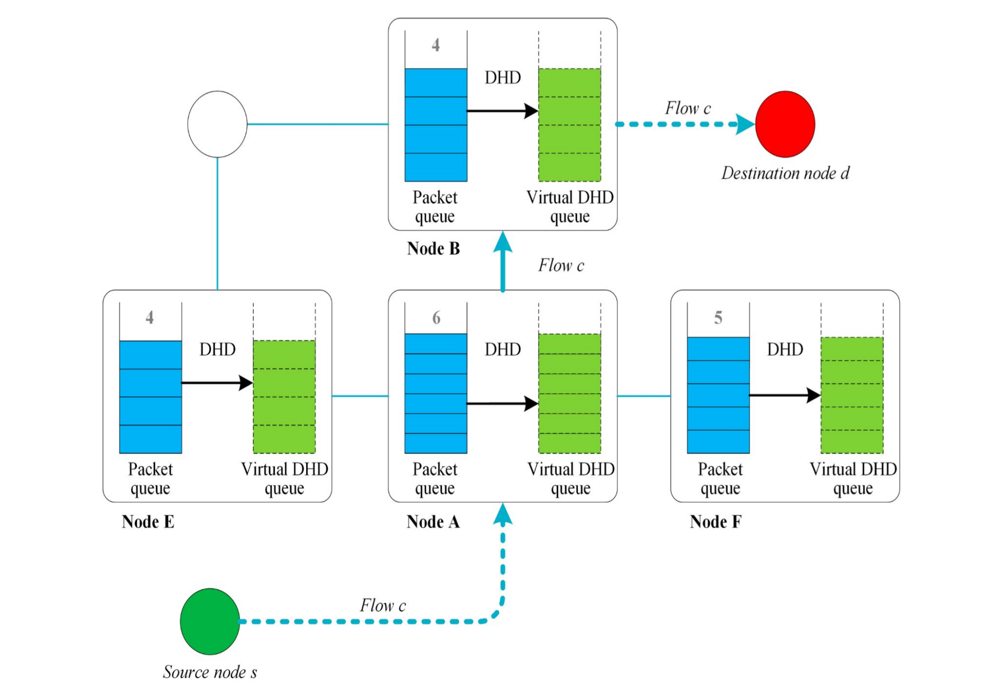
Distributed Hops Based Back-Pressure Routing

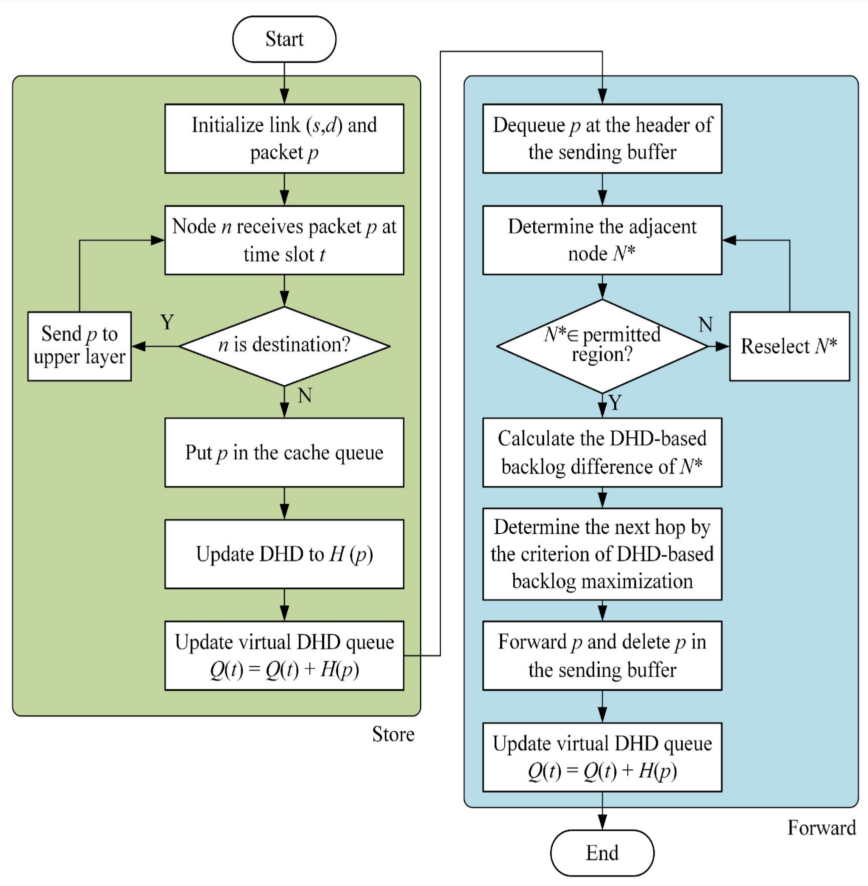
$$\hat{Q}_a^c = \sum_{p \in Q_a^c} H(p) = |Q_a^c(t)| \times H(p)$$

$$\hat{\omega}_{ab}^c(t) = \max_c \left[\hat{Q}_a^c(t) - \hat{Q}_b^c(t)\right]$$

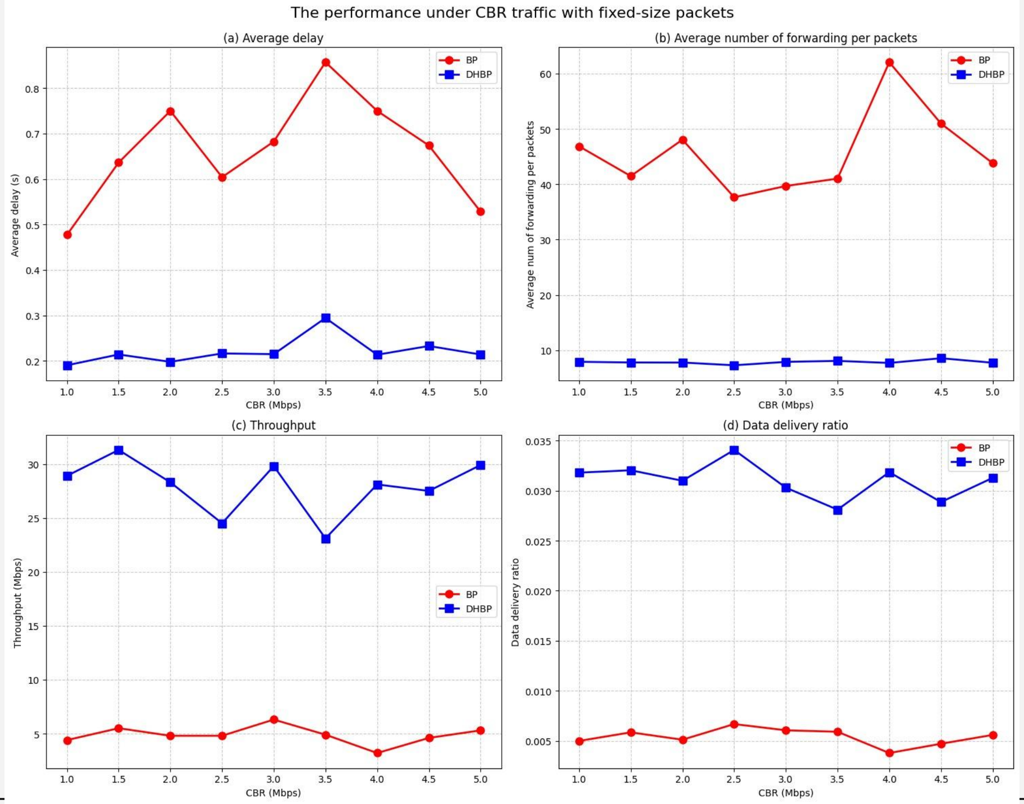
$$Maximize: \sum_{a=1}^{N} \sum_{b=1}^{N} \mu_{ab}(t) \tilde{\omega}_{ab}^c(t)$$

$$\text{s.t.} \quad \mu_{ab}(t) \in \Gamma_s(t)$$

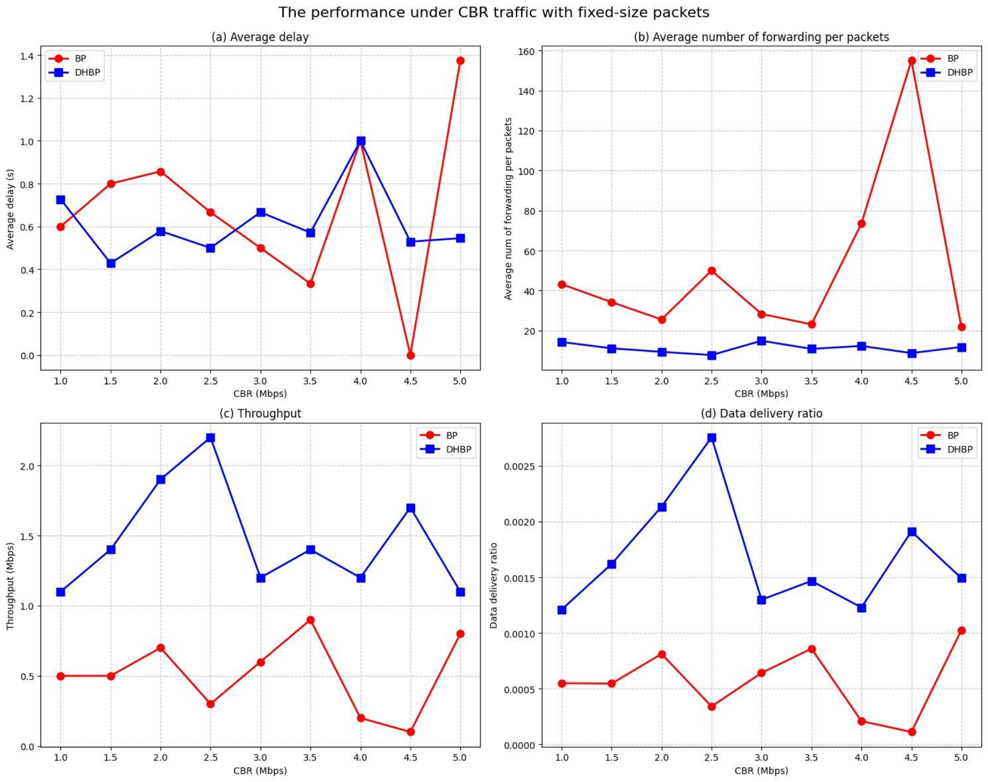



Start
Initialize link (s,d) and packet p
Node n receives packet p at time slot t
n is destination?
Y
Send p to upper layer
N
Put p in the cache queue
Update DHD to $H(p)$
Update virtual DHD queue $Q(t) = Q(t) + H(p)$
Store
Dequeue p at the header of the sending buffer
Determine the adjacent node N^*
$N^* \in$ permitted region?
N
Reselect N^*
Y
Calculate the DHD-based backlog difference of N^*
Determine the next hop by the criterion of DHD-based backlog maximization
Forward p and delete p in the sending buffer
Update virtual DHD queue $Q(t) = Q(t) + H(p)$
Forward
End

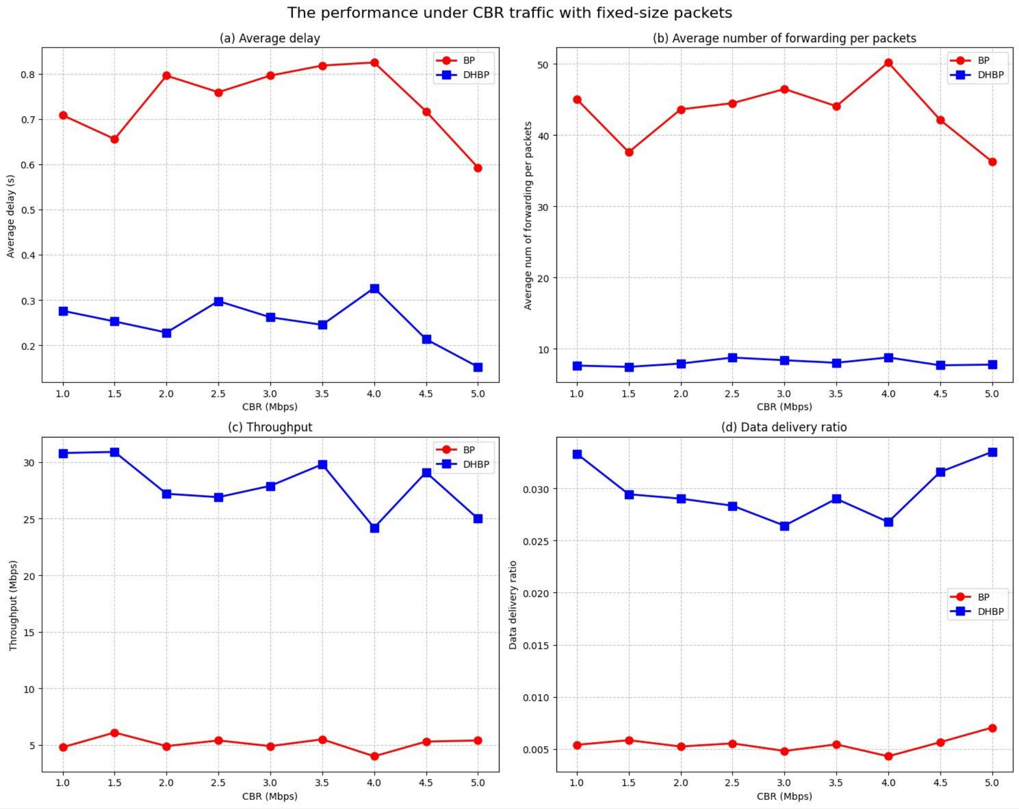
Results / recreated base paper results



Results / base paper includes broadcast mechanism



Results / Final Results With Arima



Conclusion

- We are getting better results by integrating QoS in HBPR Algorithm.

References

- https://drive.google.com/drive/folders/1D9SD-eXOwh5KzE8oAoiG-zBAXMAhag1T?usp=sharing