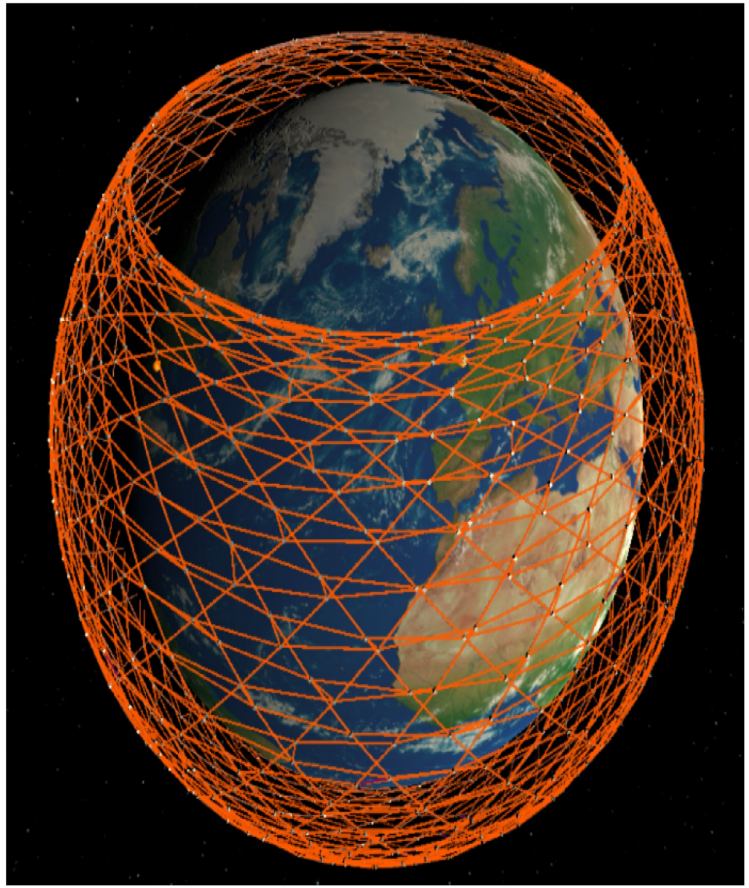
Project report-1

Project Report-1

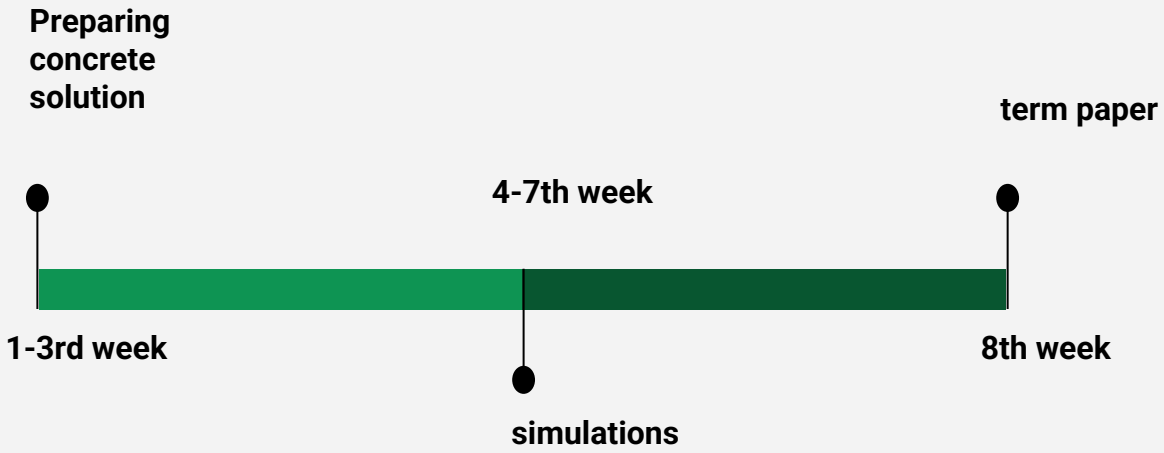
HBPR Based QoS Routing in LEO Satellites

Presented by
Sumanth Guptha M & Sasank

Mentored by Rahul Agrawal



Timeline of our project



Preparing Concrete Solutions

Our ideas :

- Applying multiple queuing for the satellites.

- Adding alpha factor for the for the satellites.

Applying multiple queuing

Q)what is multiple queuing?

- A system where tasks, packets, or processes are classified into separate queues based on priority, type, or congestion level.
- This improves traffic management, QoS, and load balancing by dynamically scheduling resources across different queues.
- But due to this there are many disadvantages.

Pros and Cons of Applying Multiple queuing

- It prioritizes the critical traffic and leaves the lower traffic then it may leads to the starvation.
- It requires complex stability analysis to ensure fairness but it can maintain bounded queue lengths.
- It works well in large-scale LEO networks but it requires more complex queue management.
- It provides more flexibility of different service levels but it requires fine-tuning of queue priorities to prevent unfair bandwidth allocation.

Adding Alpha Factor to the present Equations

Multiple ways of adding Alpha Factor

→ Multiplicative $\alpha^c Q^c$

→ Additive $Q^c + \alpha^c P^c$

→ Normalised $\omega_{ab}^c(t) = (1 - \alpha^c) \cdot \frac{Q_a^c(t) - Q_b^c(t)}{\max(Q)} + \alpha^c \cdot \frac{P^c}{\max(P)}$

Stability Analysis

Lyapunov function :

$$L(t) = \sum_{c \in C} \sum_{n \in N} Q_n^c(t)^2$$

Lyapunov drift :

$$\Delta L(t) \leq M + \sum_{c \in C} \sum_{n \in N} Q_n^c(t) \left[\lambda_n^c - \sum_{b \in N} \mu_{nb}^c + \sum_{a \in N} \mu_{an}^c \right]$$

Bounded queue stability conditions :

$$\lambda_n^c - X_n^c + Y_n^c \leq 0$$

Stability Analysis

Normalized Equation :

$$\sum_{c\in C}\sum_{b\in N}\sum_{n\in N}\mu_{nb}^{c}\left[\frac{Q_{n}^{c}(t)-Q_{b}^{c}(t)}{\max(Q)}+\alpha^{c}\frac{P^{c}}{\max(P)}\right]$$

Substituting with Lyapunov drift :

$$\Delta L(t)\leq M+\sum_{c\in C}\sum_{n\in N}Q_{n}^{c}(t)\left[\lambda_{n}^{c}-\sum_{b\in N}\mu_{nb}^{c}+\sum_{a\in N}\mu_{an}^{c}\right]$$

Expanding with ω_{ab}^{c}**:**

$$\Delta L(t)\leq M+\sum_{c\in C}\sum_{n\in N}Q_{n}^{c}(t)\left[\lambda_{n}^{c}-\sum_{b\in N}\mu_{nb}^{c}+\sum_{a\in N}\mu_{an}^{c}\right]+\sum_{c\in C}\sum_{b\in N}\sum_{n\in N}\mu_{nb}^{c}\left[\frac{Q_{n}^{c}(t)-Q_{b}^{c}(t)}{\max(Q)}+\alpha^{c}\frac{P^{c}}{\max(P)}\right]$$

Comparison between various Methods of Alpha Factor

Factor	Multiplicative $\alpha^c Q^c$	Additive $Q^c + \alpha^c P^c$	Normalized $(1-\alpha^c)\cdot\frac{Q_a^c(t)-Q_b^c(t)}{\max(Q)}+\alpha^c\cdot\frac{P^c}{\max(P)}$
Effect on Stability	Can amplify congestion differences	More balanced and stable	**Most stable**
Low-Priority Flow Impact	High risk of starvation	Ensures fair access	**Fair for all flows**
High-Priority Flow Handling	Strict prioritization	Controlled prioritization	**Smooth QoS prioritization**
Queue Growth	Risk of excessive queue buildup	Bounded queue growth	**Fully bounded growth**
Implementation Complexity	Similar to standard BP	Similar to standard BP	**Same as standard BP**

Comparison between Multiple Queues Vs Single Queue with Alpha Factor

Factor	Multiple Queues	Single Queue with Alpha Factor
Computational Overhead	High (Multiple queues, complex scheduling)	Low (Single queue, simple scheduling)
Starvation	Possible (lower-priority queues may get starved)	Less likely (dynamic priority balancing)
Single Overhead	High (sync & load balancing issues)	Low (unified processing, minimal sync overhead)

Whats Next?

→ Deciding on the Value of Alpha α^c

Fixed (Static) Values (trails based)

Adaptive (Dynamic) Tuning Based on Network Conditions

Delay-Based Adjustment

$$\alpha^c = \frac{\text{Target Delay}^c}{\text{Measured Delay}^c}$$

→ Convergence analysis over the time

→ Building the Simulation