Hex Code

SMART INDIA HACKATHON 2024

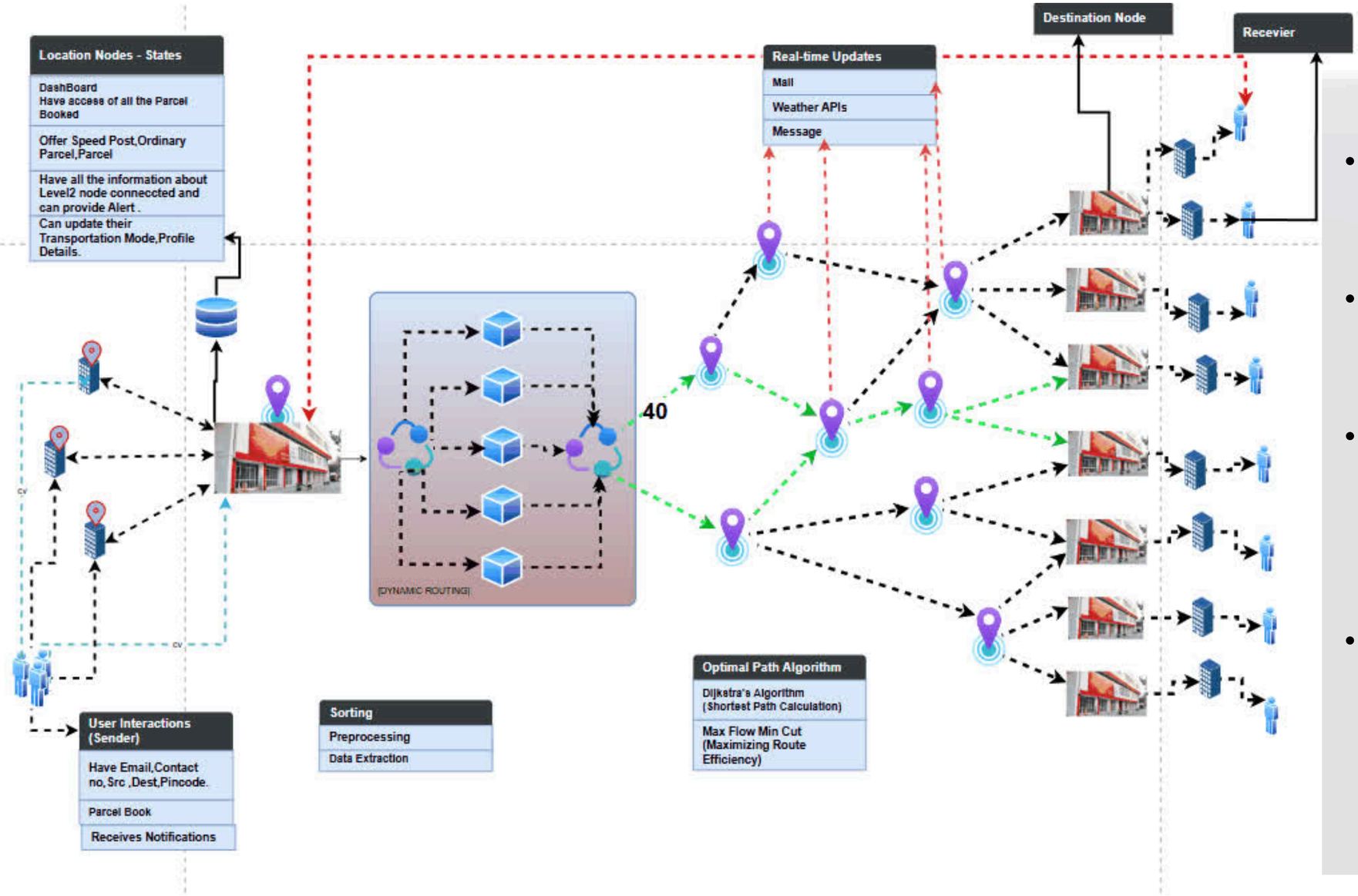
Dynamic Mail Transmission Solution

- **Problem Statement ID - 1759**
- **PROBLEM STATEENT TITLE - *Dynamic mail transmission solution using best connectivity across modes***
- **Theme - Transportation & Logistics**
- **PS Category - Software**
- **Team ID - 46401**
- **Team Name - HexaCode**

Hex Code

PROPOSED SOLUTIONS

SMART INDIA HACKATHON 2024



- **Modes of Transportation**
  - state to state(Flights, Ships, Trains, Trucks)
  - state to city (Trains,Trucks,(Flights and Ships in very few connections))
- **Delivery Process**
  - Post/Parcel send to state/city
  - Last state transferred to city
  - The last city delivers to the receiver's address.
- **Timely alerts**
  - Alert to the customer(sender and receiver) regarding reason for delay and expected delay.
  - Alert the receiver and sender's post-office regarding expected delay, reason for delay and also enabling inter node communication.
- **Customer Options to Choose From (Delivery Types)**
  - Speedpost: Delivery within 3 days
  - Ordinary post
  - Parcel delivery

Hex Code

FEATURES INCLUDED

SMART INDIA HACKATHON 2024

- Dynamic Routing based on Space, Time, Size, Fragility
- Prediction of Delivery Date and Time
- Tracking of Parcels
- Real Time Notifications/Alerts
- New Node Registration (State and City)
- Customer Support and AI Chatbot
- User Friendly Interface (Light and Dark Mode)

1. Predicting optimum mode of routing and transmission using the best available connectivity, considering real-time data and availability of space.
2. Mode of transmission that will be interactive with the transport agency.
3. The solution should also have interface for monitoring mail transmission, suggesting alternative routes, self-learn from previous situations to identify the fastest mode of transmission utilizing all available channels.
4. Alerts to the customer (both sender and addressee) may be sent conveying information about any changes/delay in normal transmission caused by any beyond-control situations.

BACKEND

Algorithm

Algorithm for Delivery Prioritization

Input

1. **Directed Graph G=(V,E):**
   - **V: Set of nodes.**
   - **E: Set of edges, where each edge(u,v) is characterized by:**
     - **d(u,v): Departure time.**
     - **a(u,v): Arrival time.**
     - **c(u,v): Capacity.**
     - **cost(u,v): Cost of traversal.**
2. **Parcels P:**
   - **Each parcel pi is defined by:**
     - **si: Source node.**
     - **ti: Destination node.**
     - **wi: Weight of the parcel.**
     - **modei: Delivery type (speed post, normal post, or common parcel).**

h

Objective

- **Minimize the overall delivery time for all parcels, prioritizing delivery by mode.**
- **Respect capacity constraints f(u,v)≤c(u,v)on edges.**

BACKEND

Algorithm

Steps:

Step 1: Sort Parcels by Delivery Mode

- **Divide parcels P into three categories:**
  - **Speed post (Pspeed).**
  - **Normal post (Pnormal).**
  - **Common parcel (Pcommon).**
- **Sort the parcels in Pspeed,Pnormal,Pcommon by their departure times.**

Step 2: Pathfinding with Mode Prioritization

- **For each category Pmode∈{Pspeed,Pnormal,Pcommon}**
  - **For each parcel pi∈Pmode:**
    1. **Find Feasible Paths:**
       - **Use a time-aware Dijkstra's algorithm to find the path πi from ss to ti, such that:**
         - **The path respects the departure time d(u,v).**
         - **All edges along the path have sufficient residual capacity r(u,v)=c(u,v)−f(u,v)**
    2. **Update Flow and Timing:**
       - **For each edge (u,v) in the path πi, update:**
         - **f(u,v)=f(u,v)+wi(increase the flow).**
         - **If f(u,v)>c(u,v), reject this path and try the next available edge.**
    3. **Mark Parcel as Delivered:**
       - **If a valid path is found, record the delivery sequence and its arrival time T[ti].**
       - **If no valid path is available, mark pi as undeliverable.**

BACKEND

Algorithm

Step 3: Combine Paths and Sequence

- **After processing all categories:**
  a. **Combine the delivery paths:**
     - **Start with Pspeed, followed by Pnormal, and then Pcommon.**
  b. **Resolve conflicts:**
     - **If multiple parcels share edges, prioritize higher-priority parcels (speed > normal > common).**
  c. **Ensure all deliveries respect time and capacity constraints.**

Mathematical Representation

1. **Residual Capacity:**
2. **r(u,v)=c(u,v)−f(u,v),∀(u,v)∈E.**
3. **Time Feasibility:**
   - **tcurrent≥d(u,v)(departure time satisfied).**
   - **Path arrival time:**
   - **T[v]=min(T[v],T[u]+(a(u,v)−d(u,v))).**
4. **Flow Update:**
   - **For each edge (u,v) in the path πi\pi_iπi: f(u,v)=f(u,v)+wi,f(u,v)≤c(u,v).**
5. **Delivery Order:**
   - **Prioritize Pspeed>Pnormal>PcommonP_**
   - **Combine paths to maximize delivery success.**

BACKEND

Algorithm

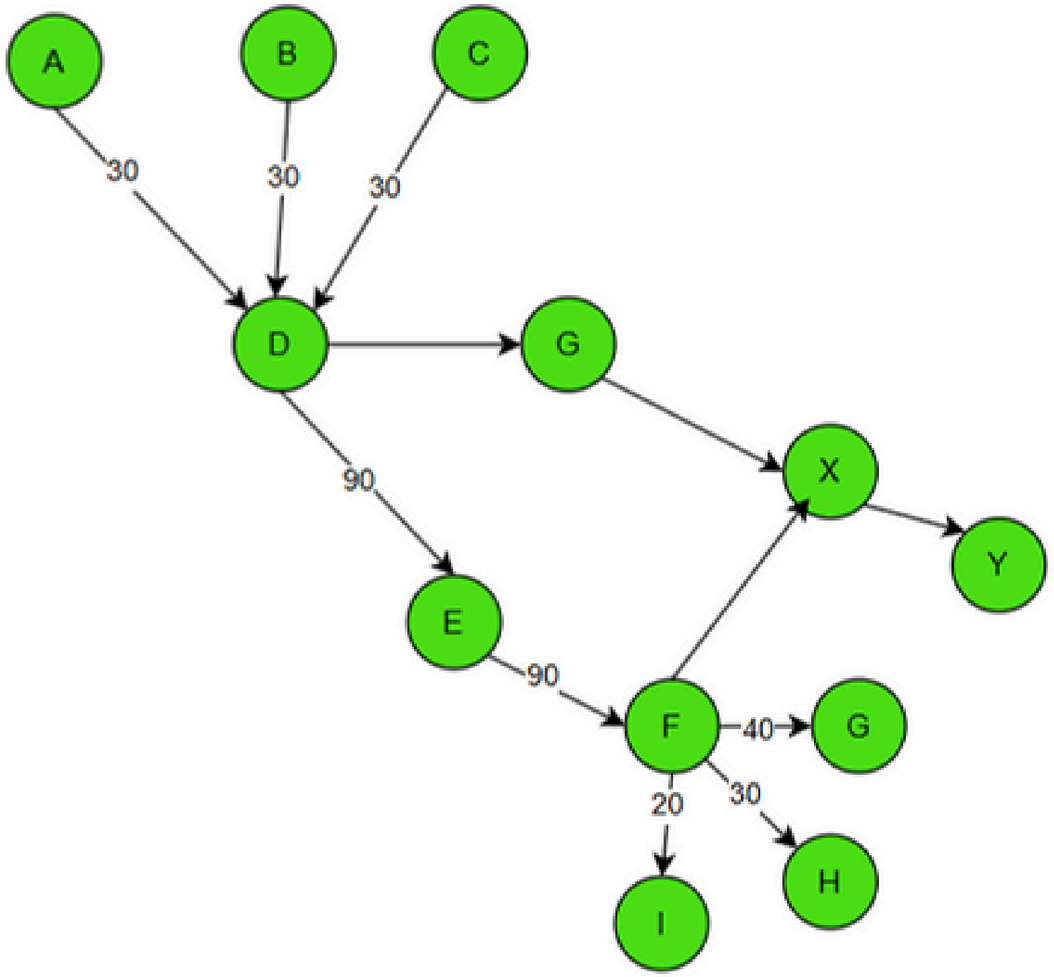
Output

1. **Delivery Schedule:**
   - **For each parcel pi, provide:**
     - **Delivery path πi.**
     - **Arrival time T[ti].**
2. **Undeliverable Parcels:**
   - **Report parcels that cannot be delivered due to constraints.**

Hex Code

BACKEND

Algorithm



10	A	I
20	A	H
30	B	G
10	c	I
10	C	G
10	C	H

SMART INDIA HACKATHON 2024

FUTURE SCOPE AND FURTHER IMPROVEMENTS

Real-Time Monitoring of Weather Conditions

Integrating Crowd Shipping

Real-time monitoring Traffic Status