Each Vehicle Thread | Intersection Manager

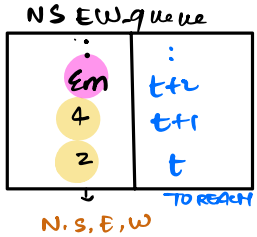
Storage Zone

When in acceptable radius:
and IM detects the vehicle presence

- Sends REQUESTS
- Update t every few sec
  OR
  Acceleration data?
- CANCEL changed requests
  (car behind takes its place)

1

- Recieve REQUESTS
- Arranges in First Ready Out
- Force CANCEL late REQUEST



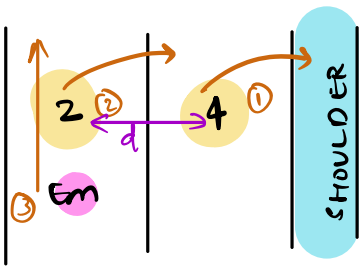
Late: Stopped sending data

2

Prioritization (EV detected)

Vehicles recieving YIELD:

- Deaccelerates, move to shoulder



Calculate lost time of vehicles moved

d = heversine formula (latitude, longitude)

t_{lost} : d/vel.

MOVED_REQUEST w/ new time

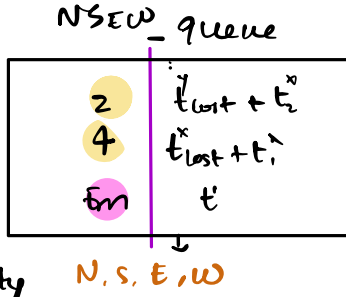
4

- Send YIELD signal to veh. in front of EV
- YIELD contains direction to be moved.

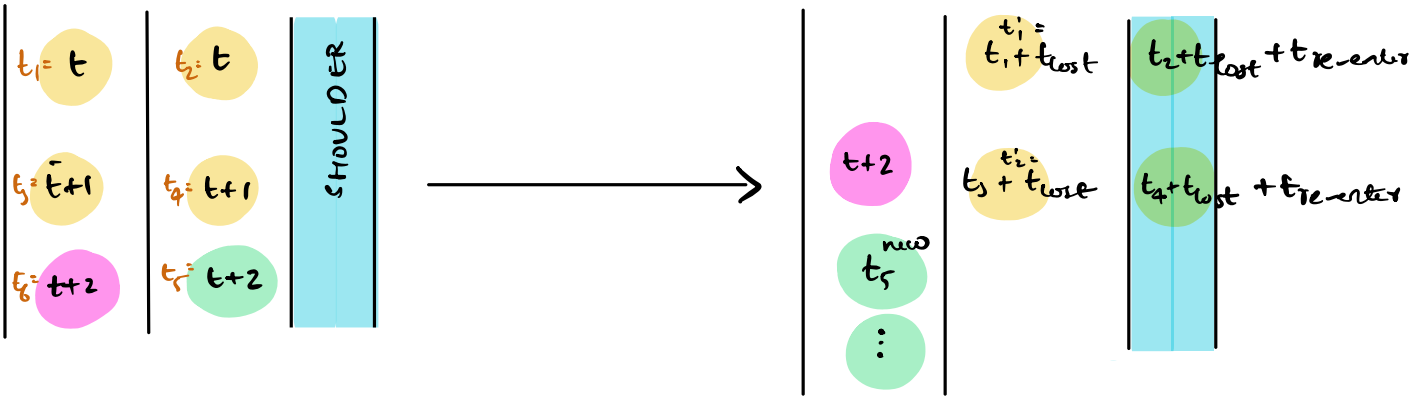
3

- CONFIRM to Em as all nodes are empty in their direction

Based on ID



MOVED_REQUEST w/ new time



t_5 moves to EV lane

$t_5^{new} = t_{emergency} + t_{renter}$

⋮

upto # of vehicles moved due to EV

$t_{shoulder} = t_{old} + t_{lost} + t_{renter}$

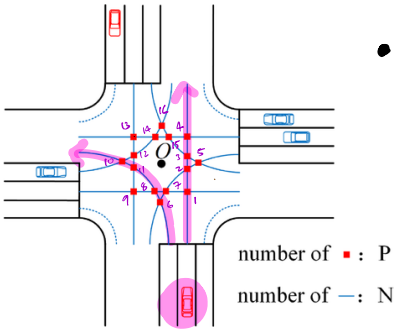
$t_{renter} = t'_{adjacent}$

$t_{right\ lane} = t_{old} + t_{lost}$

$t_{emergency} = t_{old}$

6 Vehicle Reached intersection

CONFLICT ZONE



- EM goes only 1 ways

Vehicles pass nodes at n seconds / node

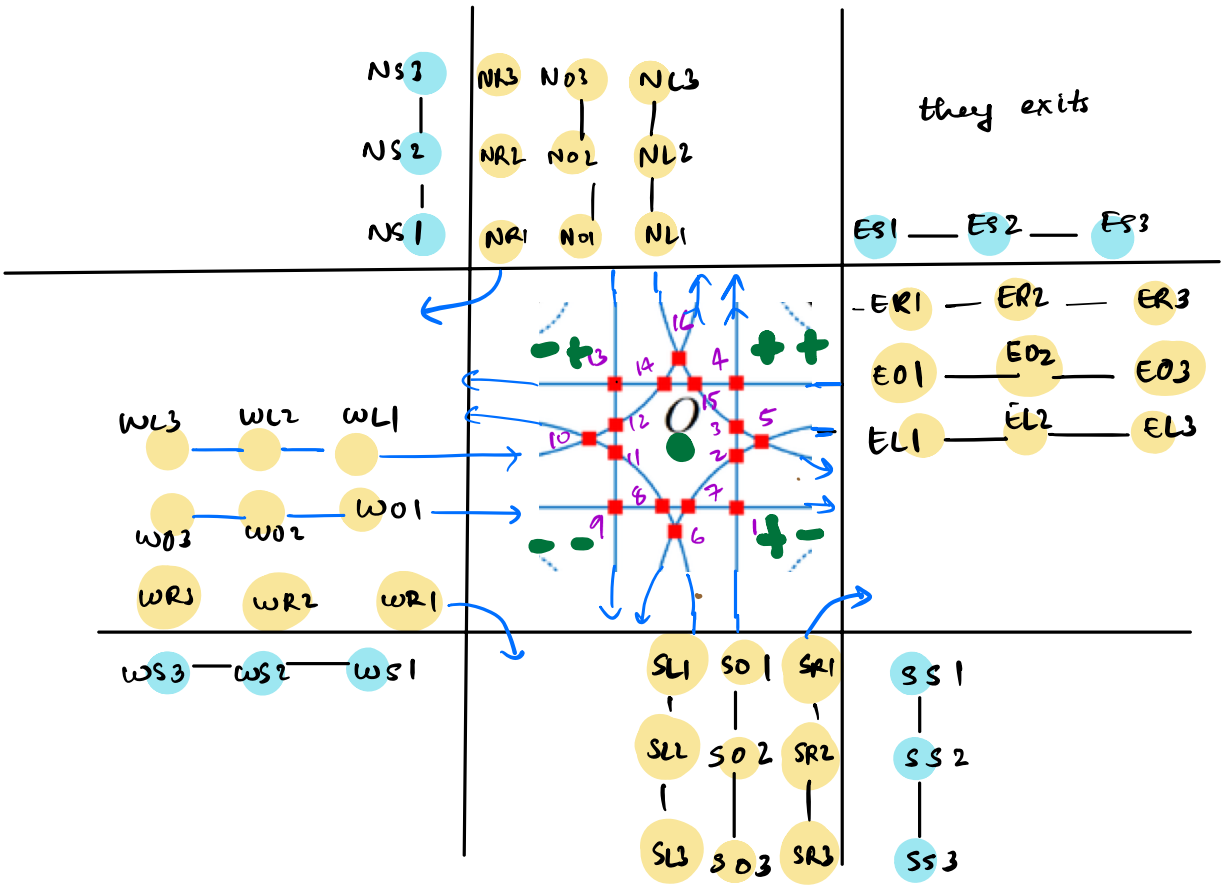
- IM blocks EV routes
- Deque REQUESTs to all directions N_S, N_E .....
- Update vehicle presence in nodes

EXIT ZONE arrays

Vehicles sent EXITED msg

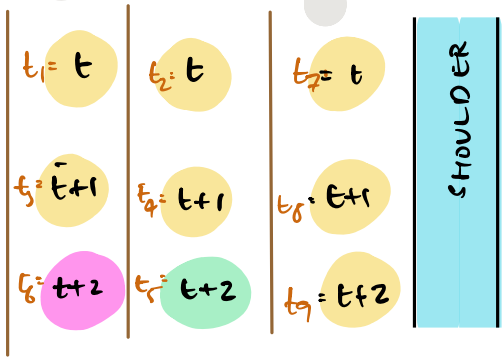
---

3 lanes :

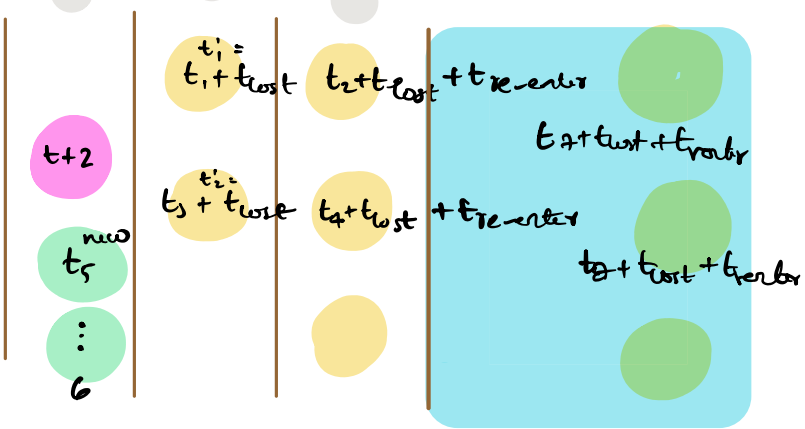


while (true) : ⟶ Clock / timer more efficient.

{ intersection manager



⟶



t_5 moves to EV lane

$t_5^{new} = t_{emergency} + t_{reenter}$

⋮

upto # of vehicles moved due to EV

$t_{shoulder} = t_{old} + t_{lost} + t_{reenter}$

$t_{reenter} = t'_{adjacent}$.

$t_{right_lane} = t_{old} + t_{lost}$

$t_{emergency} = t_{old}$.