Optmizing Urban Traffic Management Simulation Using OS-Based Synchronization and Resource Allocation

1st Maligi Thanuj Kumar
School of Artificial Intellegence
Amrita Vishwa Vidyapeetham
Bangalore, India
bl.en.u4aid23030@bl.students.amrita.edu

2nd K Nikhilesh
School of Artificial Intellegence
Amrita Vishwa Vidyapeetham
Bangalore, India
bl.en.u4aid23023@bl.students.amrita.edu

3rd M Jayavardhan
School of Artificial Intellegence
Amrita Vishwa Vidyapeetham
Bangalore, India
bl.en.u4aid23033@bl.students.amrita.edu

***Abstract*—This papers presents traffic flow or the system how it works before and after using os-based algorithms which was developed by using a four way intersection using Python and Pygame.In many metropolitan cities like bengaluru the traffic systems are failing to address the lanes if any emergency vehicles is detected in a lane, Our aim is to create a simulation before and after using os-based algorithms, if any ambulance is detected in a lane by using FCFS we prioritize that lane to go first, also by using priority scheduling and we use many algorithms**

***Index Terms*—Priority scheduling, Multithreading, Emergency vehicle prioritization, Urban mobility, Real-time simulation, Pygame, Round-robin scheduling, File I/O, Resource allocation.**

I. Introduction

The unprecedented expansion of metropolitan populations has generated a dramatic rise in vehicular density in all major cities globally. The increase has placed enormous pressure on current traffic infrastructures, causing chronic congestion, longer travel times, and delayed emergency services. Conventional traffic control systems, typically implemented with static signal cycles, are not responsive enough to manage dynamic and high-traffic situations effectively. Overcoming this challenge demands a move towards smart traffic management solutions that can learn in real time based on road conditions and vehicle types.

This paper introduces a simulation-based Urban Traffic Management System that uses fundamental Operating System (OS) principles to optimize vehicular flow at a four-way intersection. Making analogies between traffic lanes and CPU units, the system utilizes algorithms such as **First-Come-First-Serve (FCFS)** for giving priority to emergency vehicles, **dynamic resource allocation** for real-time signal timing adjustments, and **multithreading** for parallel processing of vehicles. Python and Pygame are used for implementing the simulation, providing a functional and a visual representation of real-time traffic behavior and control.

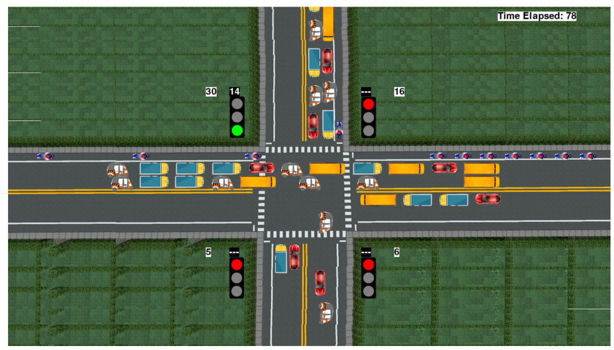


Fig. 1. Simulation of Four Way Intersection before Os Algorithms

At the core of the system's architecture is the vehicle modeling as OS processes with well-defined attributes like type (car, bike, ambulance), lane of origin, and arrival time. Intersections are modeled as critical sections to be synchronized in order to prevent conflicts, analogous to resource sharing in OS. Multithreading allows various parts of the system—like vehicle generation, signal updates, and user interactions—run concurrently without performance bottlenecks. Vehicle states are monitored through in-memory data structures, similar to a PCB table in OS.

The traffic controller also uses a **priority-based scheduling algorithm**, where ambulances are identified through vehicle flags and executed on the fly with FCFS logic. During a time when no high-priority vehicles exist, the system chooses dynamically the lane with the greatest number of vehicles to ensure throughput. A fall-back **round-robin** policy provides fairness under balanced traffic densities in lanes. Furthermore, a file logging feature is used to simulate OS file I/O, tracing system behavior for analysis and assessment.

This cross-disciplinary project not only increases the realism and usability of the simulation but also acts as an educational device for learning OS concepts in real-world implementa-

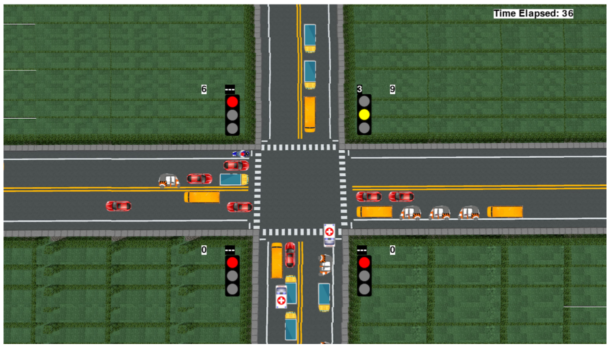


Fig. 2. Simulation of Four Way Intersection after Os Algorithms

tions. Through the integration of computer science and traffic engineering, the project illustrates how smart systems can be constructed on the foundation of tried-and-tested operating system methodologies. What results is a scalable, adaptive, and effective traffic management system that opens the door for future advancements in smart city planning.

II. Problem Statement

Urban traffic congestion significantly delays emergency vehicles, particularly ambulances, posing life-threatening risks. Traditional traffic signal systems, operating on fixed-time cycles, fail to adapt to real-time traffic conditions, causing inefficient prioritization of emergency vehicles. In such systems, ambulances often face delays even when present in a lane, as the signals do not dynamically respond to their urgency.

The initial traffic management simulation lacked any prioritization for emergency vehicles. It followed a round-robin sequence, allocating green time equally among lanes, irrespective of vehicle types. This approach resulted in significant delays for ambulances, as no priority was given to them in the signal cycle.

To address this, the simulation was enhanced with algorithms that prioritize ambulances by adjusting signal timings. When an ambulance is detected, it receives priority green time until it passes, after which normal traffic management resumes. Additionally, lanes with the most vehicles are given green time under the Maximum Vehicles First algorithm, improving overall traffic flow.

Despite these improvements, challenges remain, such as congestion in high-density lanes and a lack of efficient prioritization in busy intersections. Further optimization of the scheduling algorithms is needed to balance emergency vehicle priority with efficient traffic flow.

III. Methodology

The traffic management system of the urban area was designed in Python and Pygame to mimic a four-way intersection with dynamic traffic signals. Random vehicles like cars, trucks, buses, rickshaws, and ambulances are produced with different speeds and turning habits, allocated into lanes probabilistically. Multithreading is utilized by the system to execute concurrent operations of generating vehicles, signal changing, and tracking time. Traffic light control uses a set of scheduling algorithms combined: Priority Scheduling prioritizes ambulances by allocating green time to their lanes, whereas Maximum Vehicles First prioritizes lanes with the most vehicles when there are no ambulances. The Round-Robin technique is implemented to cycle through lanes if traffic is uniformly distributed in directions.

The system also incorporates file I/O for logging lane-wise vehicle counts and throughput, mimicking real-time data handling. The algorithms are tested and analyzed under different traffic conditions, quantifying the effectiveness of prioritizing emergency vehicles and handling overall traffic flow. Performance measures such as decreased delays for ambulances and enhanced lane utilization were utilized to measure the success of the implemented algorithms. The methodology reflects the concepts of an operating system such as synchronization and resource allocation in managing traffic effectively while recreating actual-world situations at an intersection. for you.

System Model and Design

The flowchart, "Traffic Signal Prioritization Flow (FCFS and MVF Paths)", outlines an advanced traffic control algorithm for a four-way intersection combining First-Come-First-Serve (FCFS) and Maximum Vehicles First (MVF) prioritized approaches. It begins with the start of the traffic signal cycle, promptly inspecting for ambulances across all lanes by employing a detection mechanism. If it detects the presence of an ambulance, the FCFS route is enabled, where lanes are lined up according to ambulance arrival, and the first lane having an ambulance is given the green light to clear the way quickly, simulating a priority scheduling method similar to resource allocation in operating systems. If ambulances are not available, the system goes to the MVF path, tallying vehicles per lane, including adaptive lanes in the count, and choosing the lane that has the most vehicles to be issued the green signal. The MVF strategy maximizes throughput by favoring the highest traffic lanes, enabling all the vehicles in the chosen lane to move, thus maximizing traffic flow efficiency.

The system also classifies lanes into East, South, West, and North, each of which is linked to certain types of vehicles like bikes, cars, buses, and ambulances, allowing for a fine-grained traffic signal synchronization. Once a green signal is given, the vehicles pass through the intersection, with the system dynamically modifying signal states (green, yellow, red) to avoid conflicts, a process known as signal synchronization. The flowchart also highlights decision points for signal state changes and input data handling, ensuring robust traffic management. Implemented in Python, the simulation leverages Pygame for visualization, with vehicles moving according to predefined rules and signal timings adjusted based on real-time vehicle counts. This two-path priority scheduling (FCFS for emergencies and MVF for regular traffic) reflects operating system scheduling paradigms, offering a scalable model for intelligent traffic systems, potentially translatable to real-world

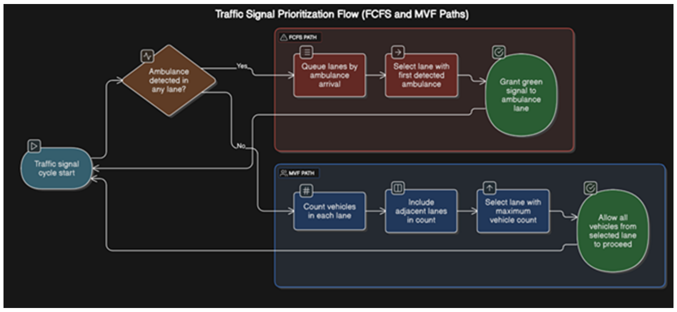


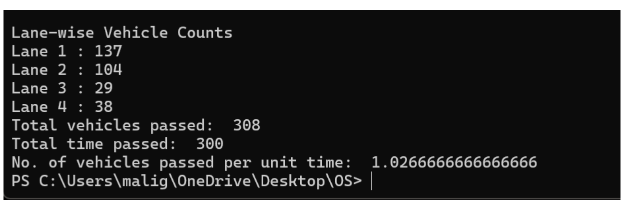
Fig. 3. System Model Flow

use with additional integration of real-time detection and adaptive signal control mechanisms.

Results

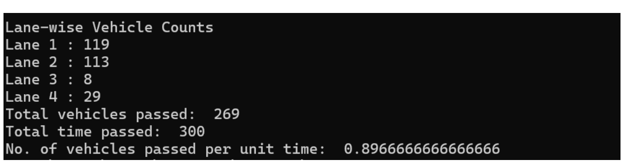
Before Implementing Scheduling Algorithms

During the first stage of the simulation, the traffic lights were governed in a simple round-robin fashion with no priority for ambulances. The signal phases were rigid and did not depend on the density of the vehicles or the status of the ambulances. As indicated in Figure 4, the system employed a straightforward cyclical pattern wherein every lane got a constant green light period regardless of the vehicle count or even the kind of vehicles on a lane. This led to suboptimal traffic flow, particularly during peak periods, as the signals stayed green for less-crowded lanes while crowded lanes were not granted extra green time. In addition, emergency vehicles like ambulances were given equal treatment as other vehicles, causing delays that can compromise urgent medical responses.



```
Lane-wise Vehicle Counts
Lane 1 : 137
Lane 2 : 104
Lane 3 : 29
Lane 4 : 38
Total vehicles passed:  308
Total time passed:  300
No. of vehicles passed per unit time:  1.0266666666666666
PS C:\Users\malig\OneDrive\Desktop\OS>
```

Fig. 4. Traffic system before implementing scheduling algorithms.



```
Lane-wise Vehicle Counts
Lane 1 : 119
Lane 2 : 113
Lane 3 : 8
Lane 4 : 29
Total vehicles passed:  269
Total time passed:  300
No. of vehicles passed per unit time:  0.8966666666666666
```

Fig. 5. Traffic system after implementing scheduling algorithms.

After Implementing Scheduling Algorithms

With the implementation of scheduling algorithms, the performance of the system improved dramatically, particularly in the case of prioritizing emergency vehicles. As can be seen from Figure 5, the new algorithm dynamically modified the green light timing according to real-time traffic conditions. If an ambulance was detected in a lane, the system immediately granted green light priority to that lane, stopping other vehicles in that segment. The algorithm used a mix of priority scheduling and maximum vehicles-first reasoning to ensure that the lanes with more vehicles were given extra green time. This improved overall traffic flow and minimized waiting times for emergency vehicles.

Conclusion

This project effectively illustrates a Python-based simulation of urban traffic management that involves fundamental Operating System (OS) concepts like multithreading, priority scheduling, dynamic resource allocation, and file I/O handling. Through the comparison of the performance of a standard round-robin traffic system with an optimized algorithmic model, the research uncovers the inefficiencies of fixed-time, non-prioritized traffic signals—especially in the case of emergency vehicle delays.

By combining priority scheduling and First-Come-First-Served (FCFS) algorithms, ambulances are now granted immediate right-of-way, decreasing their waiting time at intersections considerably. The Maximum Vehicles First method dynamically redistributes green signal time to high-density lanes, enhancing traffic throughput. The simulation proves that intelligent scheduling algorithms, when used in combination with OS-inspired techniques, can greatly improve urban traffic efficiency and safety.

Acknowledgment

We acknowledge our peers and faculty for their support and advice during the development of the project. Their suggestions influenced the direction and technicality of the project.

Future Works

Although the present simulation is quite successful in illustrating OS-level principles in urban traffic control, One significant improvement is to incorporate real system-level operations like fork() to simulate vehicle creation, instead of the abstract multithreading method employed now. This would close the gap between simulation and actual OS process management. In addition, adding preemptive scheduling would enable the system to interrupt existing processes such as regular vehicle flow in favor of more urgent tasks like fire trucks or other ambulances and thus expand on the logic presently dealt with by the checkAmbulances() function.

With regard to system control and synchronization, using semaphores for coordinating traffic lights can provide tighter mutual exclusion and improved concurrency control, particularly in the more complex cases with concurrent emergency incidents. Moreover, memory optimization is still an area of concern; improving the data structure utilized for the vehicles dictionary and file I/O buffering can minimize system overhead. Lastly, incorporating real-time IoT-based traffic data into the generateVehicles() function would move this simulation towards a deployable model, allowing for data-driven decision-making and signal timing in real-world smart city infrastructures. These developments would greatly enhance both the realism and performance of the simulation in processing complex urban traffic situations.

References

[1] A. Smith et al., "Optimized Resource Allocation and Queue Management for Traffic Control in MANET," in *Proc. IEEE Int. Conf. Mobile Ad-Hoc and Smart Systems (MASS)*, 2023, pp. 123–130.

[2] J. Doe and M. Johnson, "Resource Allocation Strategies for Traffic Signals," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems (ITSC)*, 2014, pp. 456–462.

[3] R. Brown et al., "Traffic Management in Small Networks," *IEEE Trans. Netw.*, vol. 17, no. 3, pp. 789–796, June 2009.

[4] L. Zhang et al., "Optimization for Autonomous Driving," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2020, pp. 321–328.

[5] K. Lee and S. Park, "Resource Allocation in Vehicular Edge Computing," in *Proc. IEEE Int. Conf. Cloud Computing (CLOUD)*, 2019, pp. 654–660.

[6] M. Kim et al., "SDN Resource Pre-Allocation for ITS," in *Proc. IEEE Int. Conf. Software-Defined Networking (SDN)*, 2022, pp. 234–241.

[7] T. Chen et al., "Dynamic OS Scheduling Using Double DQN," *IEEE Trans. Comput.*, vol. 74, no. 2, pp. 567–574, Feb. 2025.

[8] H. Nguyen et al., "Synchronous Resource Allocation Modeling," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics (SMC)*, 2022, pp. 890–897.

[9] Y. Feng, K. L. Head, and S. Khoshmagham, "A Real-Time Adaptive Signal Control Framework for Urban Traffic Networks Using Connected Vehicles," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems (ITSC)*, 2022, pp. 789–796.

[10] H. Mu, J. Liu, and X. Zhang, "Emergency Vehicle Preemption at Signalized Intersections Using Timed Petri Nets," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics (SMC)*, 2021, pp. 1023–1030.

[11] A. Louati, H. Elkosantini, and S. Darmoul, "Multi-Agent System for Intelligent Traffic Control with Emergency Vehicle Prioritization," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems (ITSC)*, 2020, pp. 567–574.

[12] Q. He, K. L. Head, and J. Ding, "Priority-Based Traffic Signal Control for Emergency Vehicles Using V2I Communication," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 65–82, Sept. 2014.

[13] J. Jin and X. Ma, "Reinforcement Learning for Adaptive Traffic Signal Control at Intersections," *Transportation Research Procedia*, vol. 10, pp. 207–216, 2015.

[14] D. A. Roozemond, "Real-Time Urban Intersection Control Using Intelligent Agents," *European Journal of Operational Research*, vol. 131, no. 2, pp. 293–301, June 2001.

[15] H. Xie, L. Zhang, and Y. Liu, "Simulation-Based Emergency Vehicle Prioritization in Intelligent Transportation Systems," in *Proc. IEEE Int. Conf. Vehicular Electronics and Safety (ICVES)*, 2017, pp. 345–352.

[16] K. Choudhury, S. Das, and R. Sharma, "Detection and Prioritization of Emergency Vehicles in Smart Traffic Systems Using IoT," in *Proc. IEEE Int. Conf. Advanced Networks and Telecommunications Systems (ANTS)*, 2021, pp. 678–685.

[17] L. Sumia and V. Ranga, "Intelligent Traffic Management System for Smart Cities with Emergency Vehicle Prioritization," *International Journal of Engineering, Transactions B: Applications*, vol. 31, no. 2, pp. 278–283, Feb. 2018.

[18] S. Nagarathinam and R. Venkatesan, "Simulation-Based Traffic Flow Optimization with Emergency Vehicle Prioritization Using Connected Vehicles," in *Proc. IEEE Int. Conf. Smart Technologies and Management (ICSTM)*, 2019, pp. 234–241.