Network Traffic Simulator

Designing an Extensible Traffic Management System Using Python

Julia Ruiter

Universiteit Utrecht
MSc Applied Data Science
Department of Natural Sciences

11 July 2022

Summary

Overview of Traffic Simulators

What is a Traffic Simulator?

- **Network**: infrastructure (real, virtual, etc) that allow the flow of objects or information
- **Traffic**: objects (or packets) that transverse the network
- **Simulation**: modeling the flow of traffic on the network over time

- Macroscopic: models flow and patterns
- Microscopic: models individual units

Existing Simulations

Figure: SUMO [1]

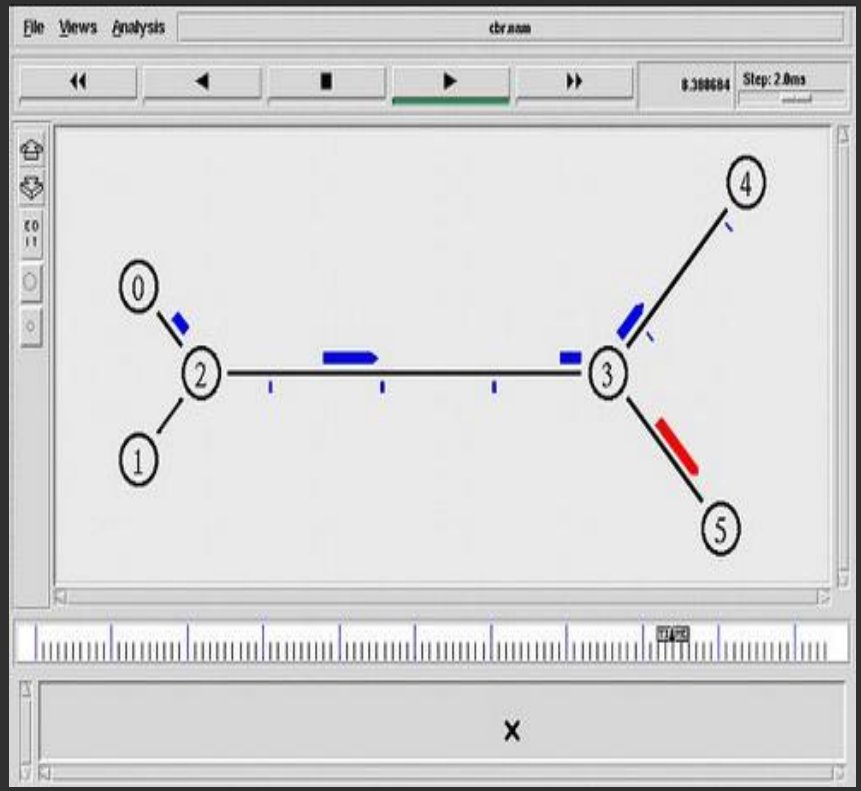


Figure: NS2 [3]

Designing a New Traffic Simulator: Architecture

Desired Network Structure

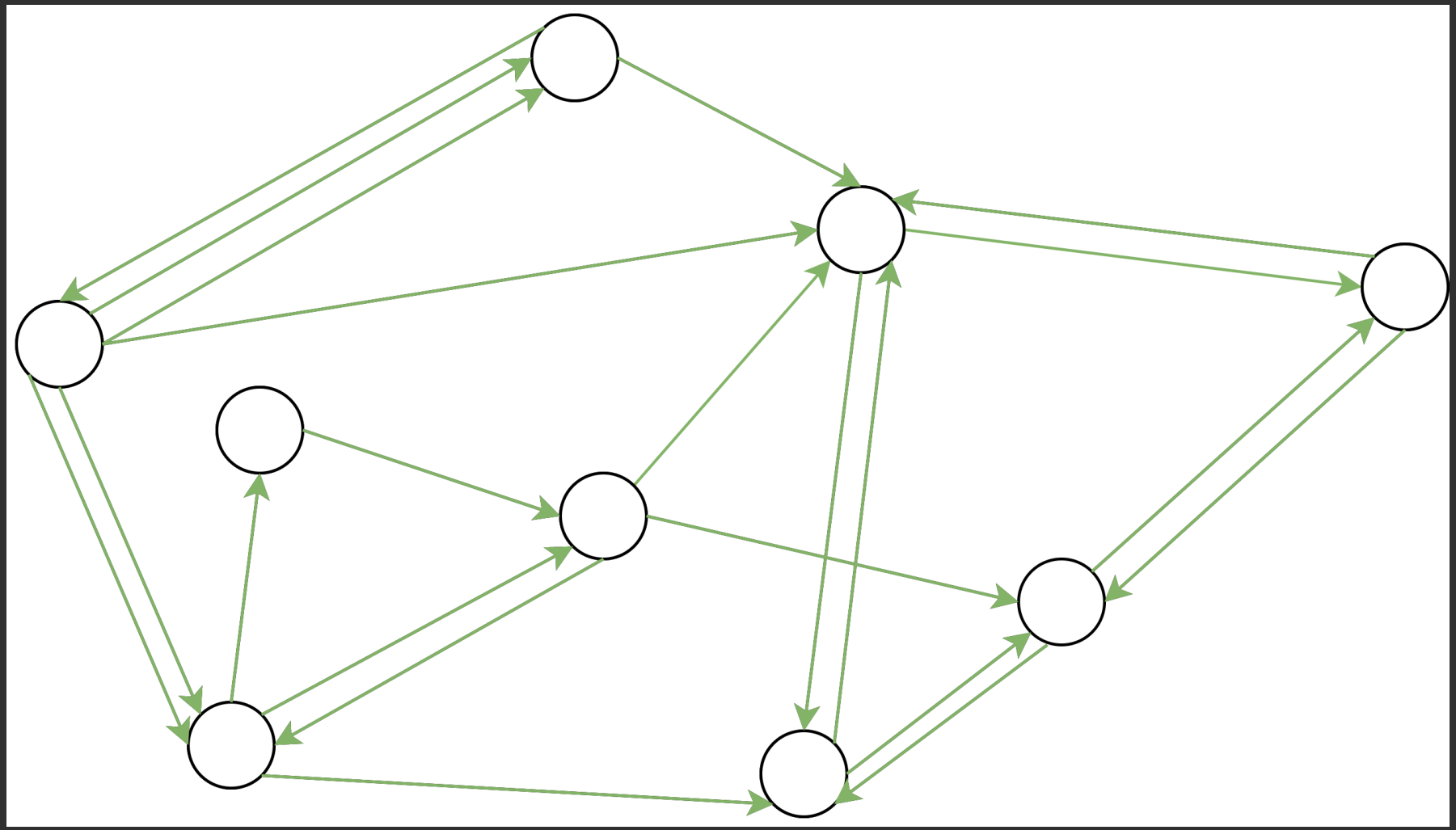


Figure: Example generalized graph network

Desired Network Structure: Highlights

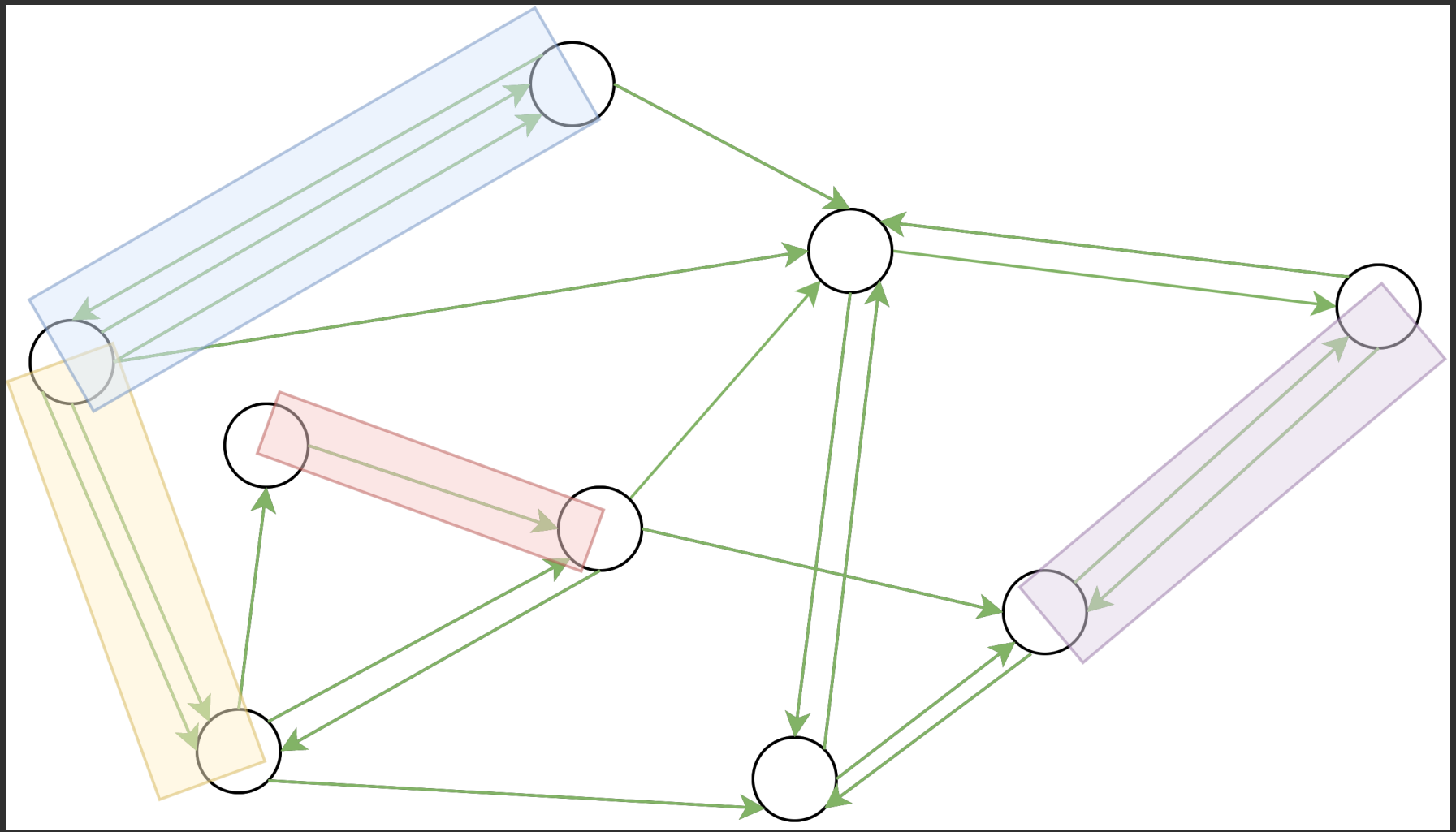
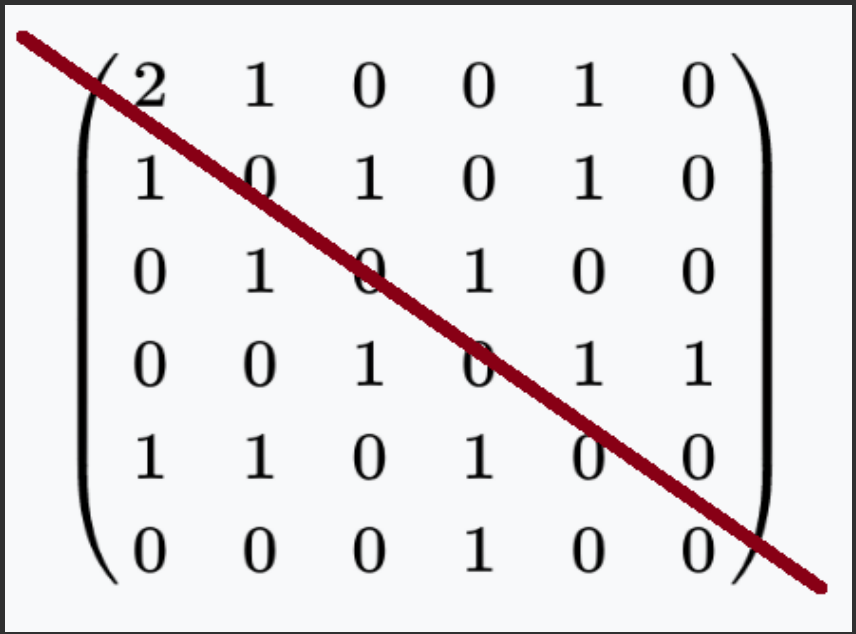


Figure: Different edge patterns have been highlighted

Desired Network Structure: Data Structures



$$\begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Figure: Adjacency matrices cannot isolate individual edges

```
defaultdict(<function Network.__init__.<locals>.<lambda> at 0x00000247F
FD2DF70>, {1: <traffic_network.Node object at 0x000002478060AD60>, 2: <
traffic_network.Node object at 0x000002478060AD90>, 3: <traffic_network
.Node object at 0x000002478060AF70>, 4: <traffic_network.Node object at
 0x000002478060AF40>})
```

```
{
    "id": 2,
    "intersection_time_cost": 0,
    "stoplight_pattern": null,
    "stoplight_pattern_current_index": 0,
    "stoplight_duration": 60,
    "stoplight_delay": 10,
    "node_tick_number": 0,
    "outbound_edges": [
        5001,
        5002
    ],
    "inbound_edges": [
        5000,
        5003
    ]
},
```

Figure: Dictionary output: internal (top), human readable (bottom)

Network Software Components

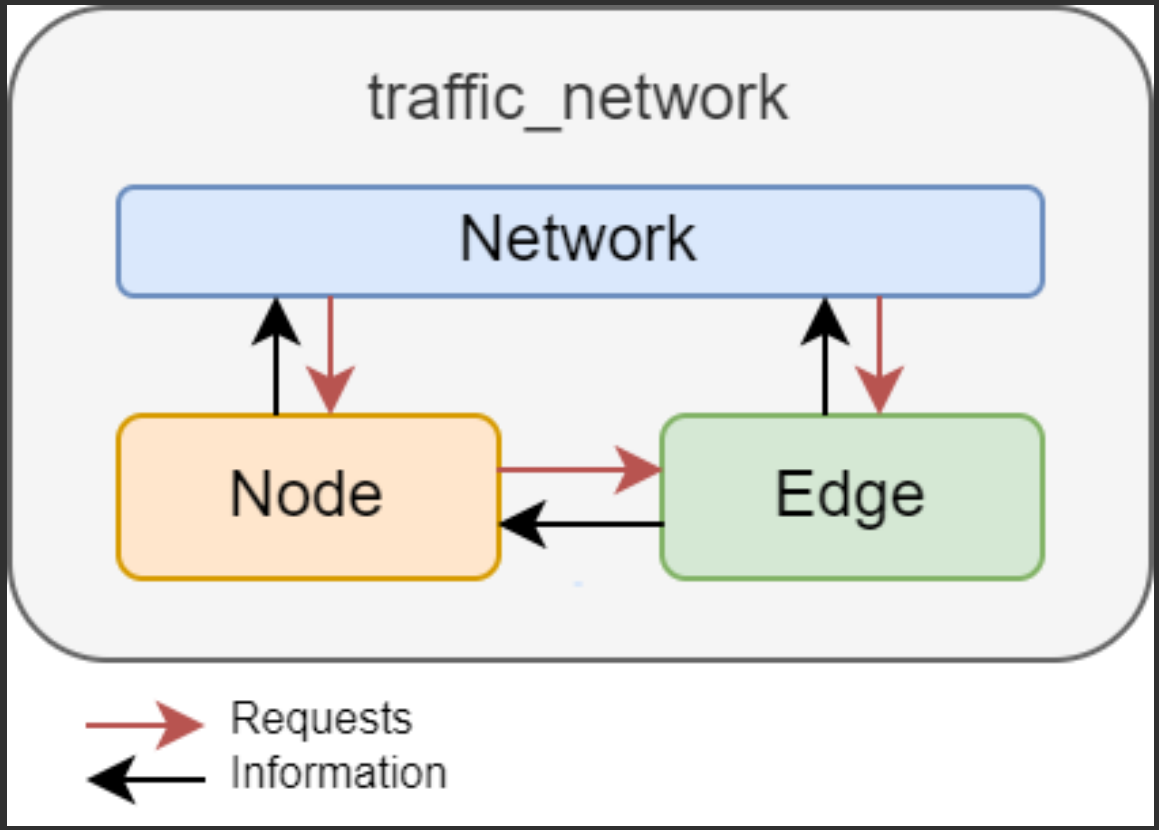


Figure: Network software module and class components

Desired Traffic Structure: Cars

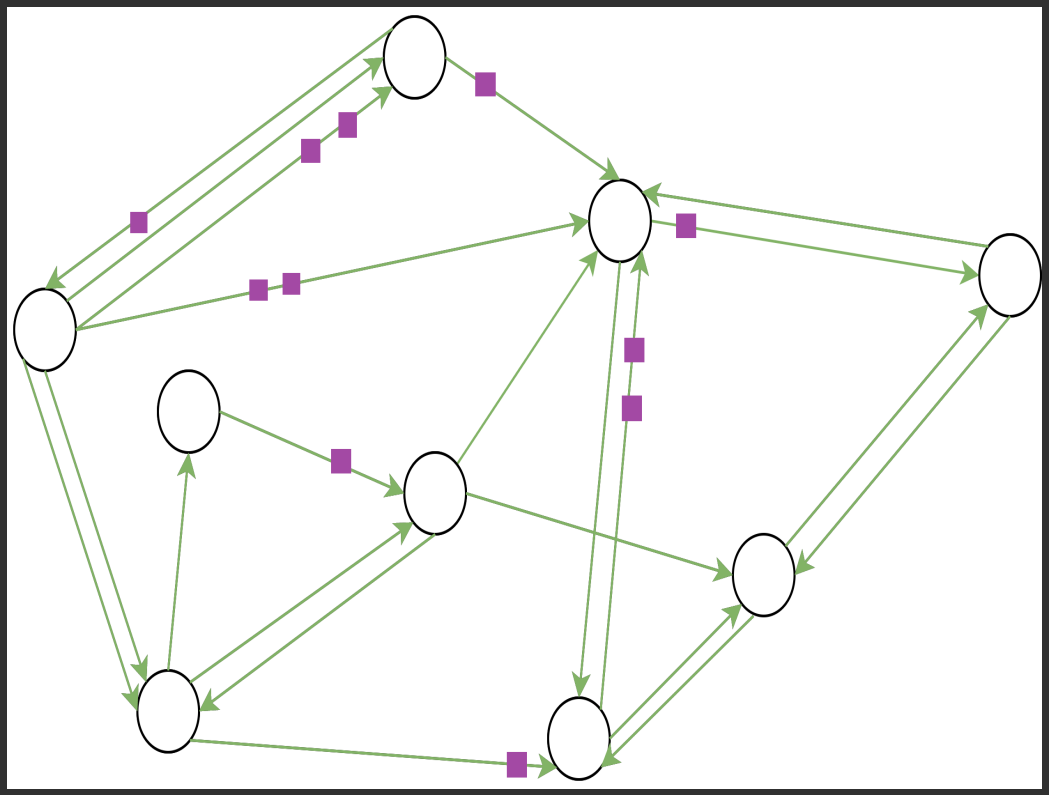


Figure: Cars (purple squares) moving along the network

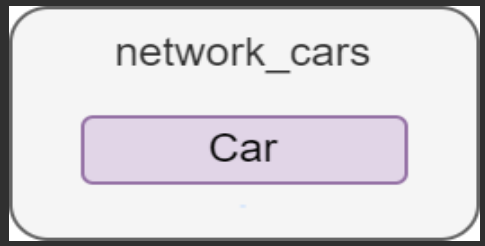


Figure: Car software module and class component

Complete Software Architecture

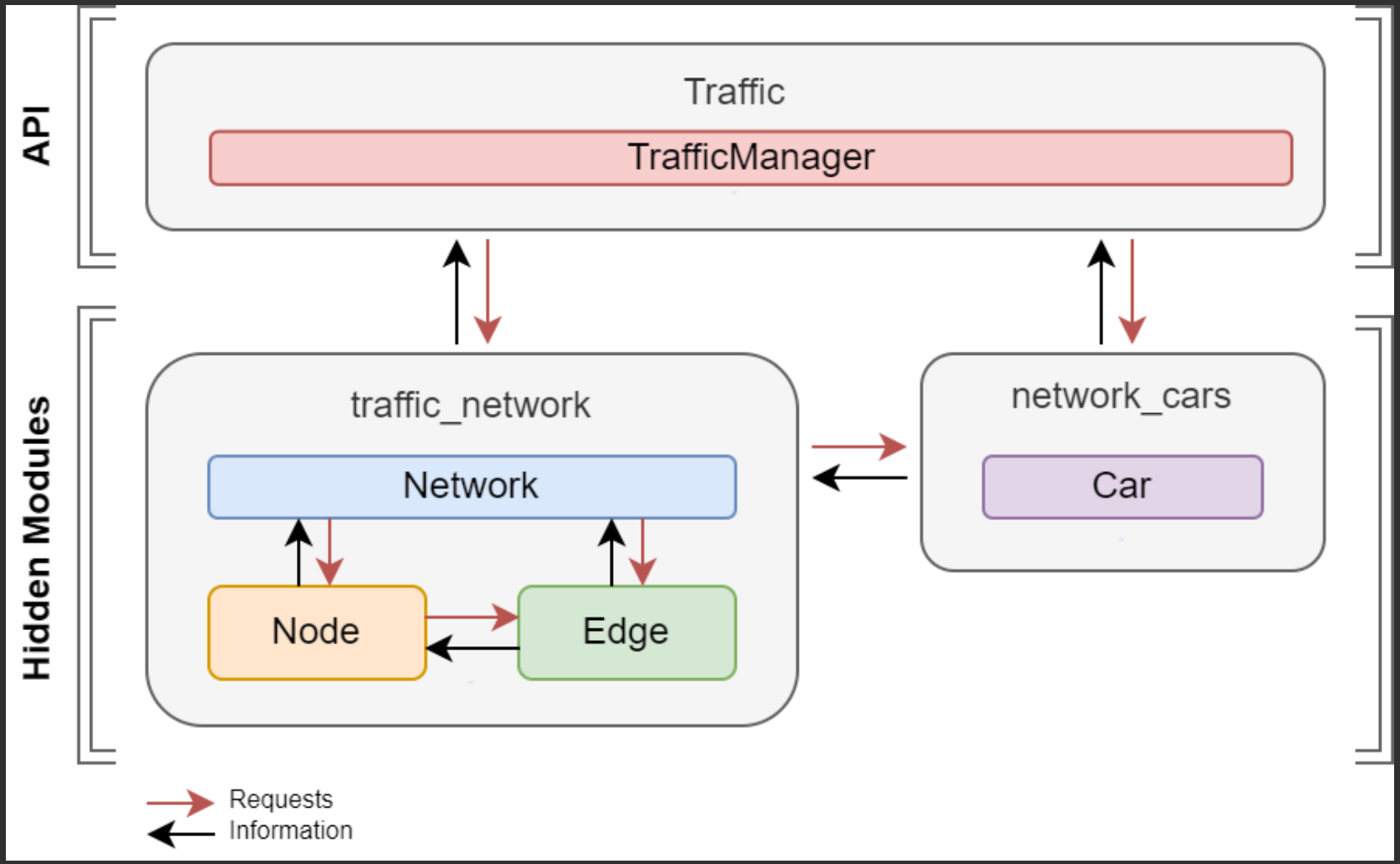


Figure: Module Interaction

Full Software Ecosystem

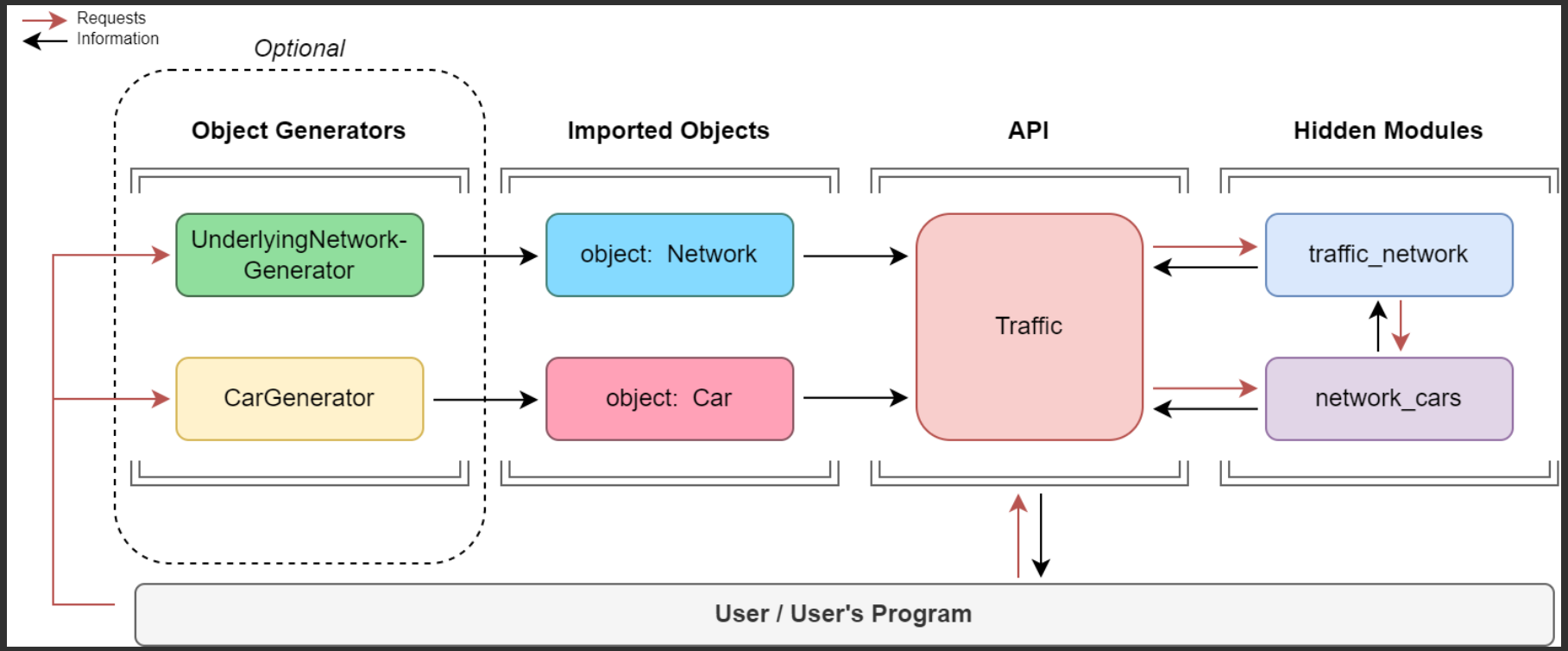


Figure: Module Interaction, including user and generators

Designing a New Traffic Simulator: Movement

Car Movement along Edges

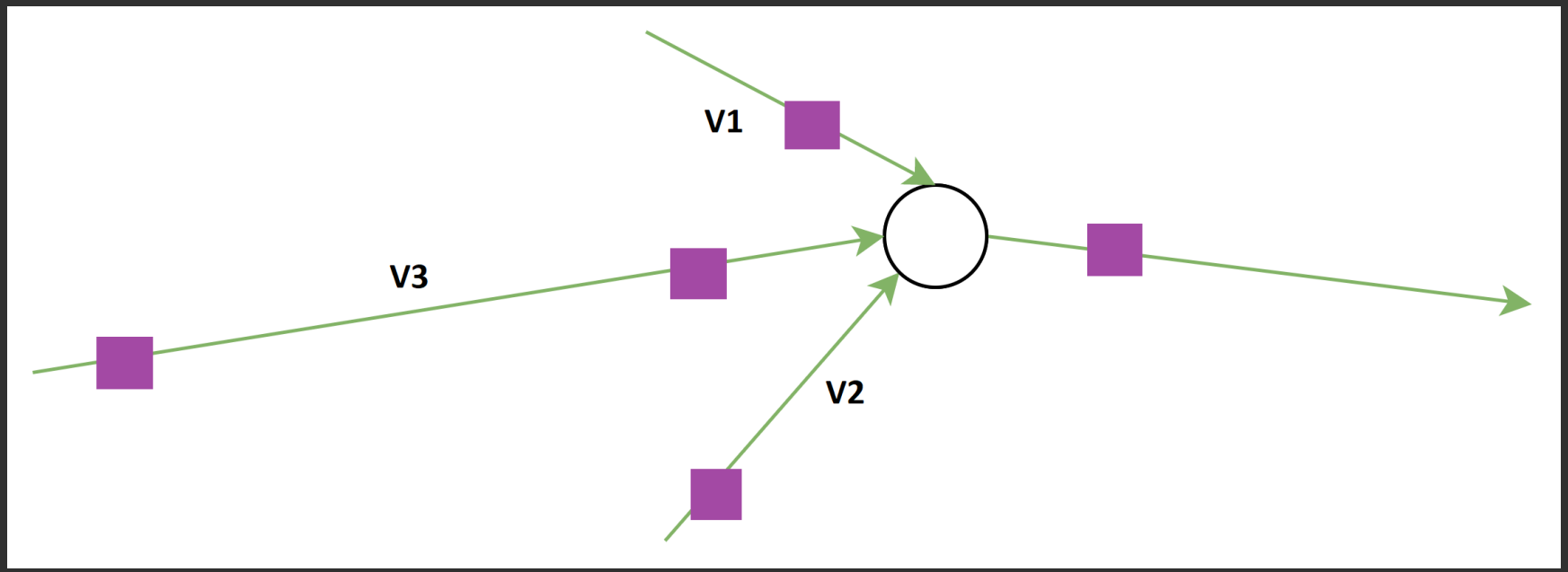


Figure: Several cars' positions at a moment in time

Movement Considerations

- Does a car have "room" to move?
- How far does a car go on to the following edge?
- What if the two edges have different speeds?
- What if cars from two (or more!) different inbound edges want to move onto the same new edge?
- Are cars allowed to change their path?

Movement Procedure

- One unit of simulation time is a tick
- The entire system ticks:
  TrafficManager $\rightarrow$ ||: Network $\rightarrow$ Node $\rightarrow$ Edge $\rightarrow$ Car :||
- Each car per tick has a maximum "potential" energy that can be utilized to move

What happens on a tick?

- **TrafficManager Tick**: tell Network to move cars as far as energy allows and advance timestamp by one tick
- **Network Tick**: tell (shuffled) Nodes to tick
- **Node Tick**: tell (shuffled) outbound edges to tick
- **Edge Tick**: tell cars to move as much as possible***
- **Car Tick**: calculate remaining potential

Node Tick and Edge Tick

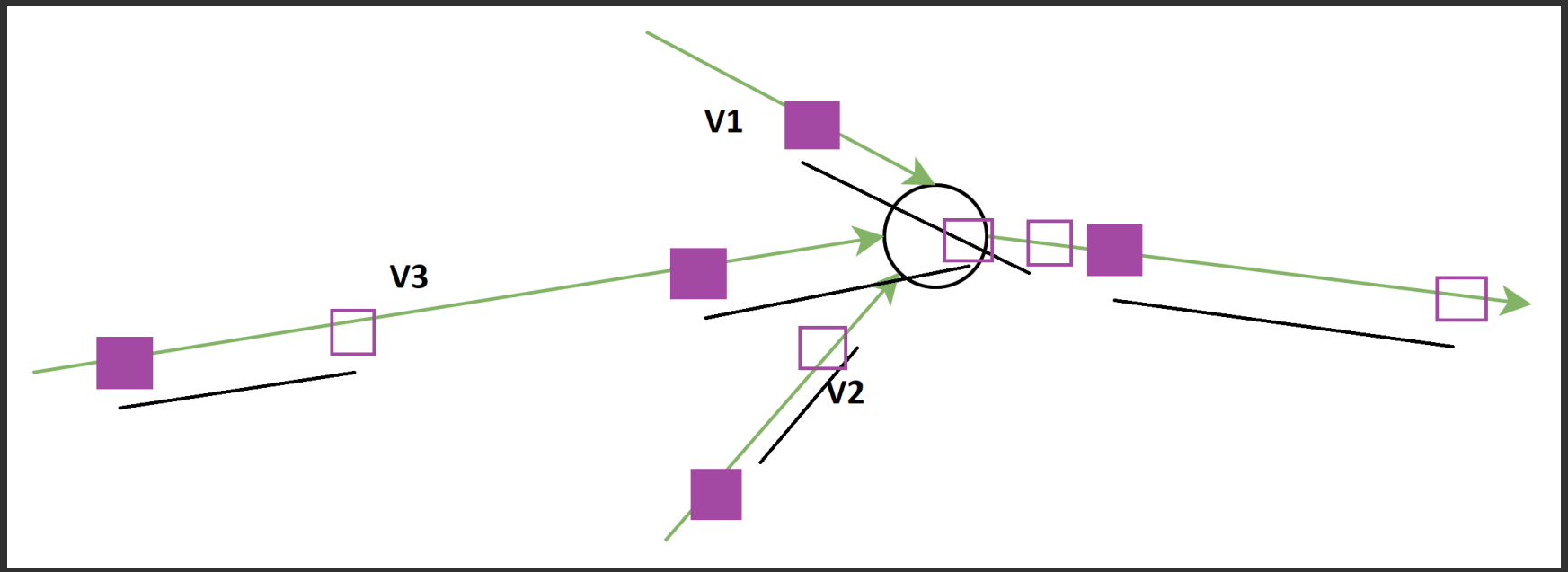


Figure: Eligible maximum travel distance per car has been mapped

Node Tick

- Check all inbound edges for cars eligible to move onto the next edge
- Place these eligible cars onto the start of the next edge in their path (if possible)
- If cars change edges, tell the old edge to forget the car and the map the car to its new edge

Edge Tick

- Try to move the car its maximum tick potential
- If it runs into another car, halt it there
- If it completes its route, remove it from the network
- If it reaches the end of the edge, wait: it will be transfered on the next node tick

Movement Procedure Recap

- The entire system ticks:
  TrafficManager $\rightarrow$ ||: Network $\rightarrow$ Node $\rightarrow$ Edge $\rightarrow$ Car :||
- Each car per tick has a maximum "potential" energy that can be utilized to move
- **If no more movement is possible on the current tick, the tick is complete. Output simulation state data.**

Next Steps

Software Improvement

- Subclasses for different car types (max tick potential, route types, etc)
- Reduce computation: flag complete and/or empty nodes and edges to be ignored on subsequent passes of the same global tick
- More efficient route calculation

User Experience / Simulation Improvement

- Create a visualizer (vis.js or D3.js)
- Add functions (delete nodes/edges)
- Congestion metrics backed by research (or user feedback)
- Snapshot improvements (logs or deltas instead of full dump)

References I

1. (DLR), G. A. C. *et al.* *SUMO User Documentation.* https://sumo.dlr.de/docs/index.html. 2022.
2. Lopez, P. A. *et al.* Microscopic Traffic Simulation using SUMO. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (2018).
3. Singh, A. *Install NS2 (Network Simulator) on Ubuntu 18.04.* https://www.absingh.com/ns2/. 2022.

Questions?