Parallel Implementation of the OMNeT++ INET Framework for V2X Communications

Ioannis Mavromatis, Andrea Tassi, Robert J. Piechocki and Andrew Nix
Dept. of Electrical and Electronic Engineering, University of Bristol, UK
{Ioan.Mavromatis, A.Tassi, R.J.Piechocki, Andy.Nix}@bristol.ac.uk

1. Problem Motivations

- Existing simulation frameworks operate in a **single thread fashion**.
- The few existing parallel features require a **laborious reconfiguration** of the existing scenarios.
- One minute of simulation time could result in **hours or days of computation**.

2. Research Questions

The necessity for faster large-scale simulations requires further investigation of the existing frameworks and ways to parallelise them:

[Q1] What are the sequential functions, related to the exchange of wireless packets, that delay the simulation time?

[Q2] How can we parallelise them in order to minimise the simulation time?

3. System Analysis

i. INET treats the positions of the vehicles as **points on the simulated canvas**.

ii. The signal propagation is modelled as **a line segment** between points A and B and is a function of:
   a. The path loss model.
   b. The obstacle loss model.

iii. INET finds in **sequential and iterative** manner all the intersections with the obstacles. For the given intersection, it calculates the obstacle loss.

iv. Due to the broadcast nature of IEEE 802.11p and vehicular networks, this generates a **complexity of** $O(mn^2)$ within our simulated scenario.

v. In city-scale scenarios, vehicles are not expected to **communicate for one side of the city to the other**.
   - Despite this, INET **computes all intersections** between two vehicles, regardless of their distance.

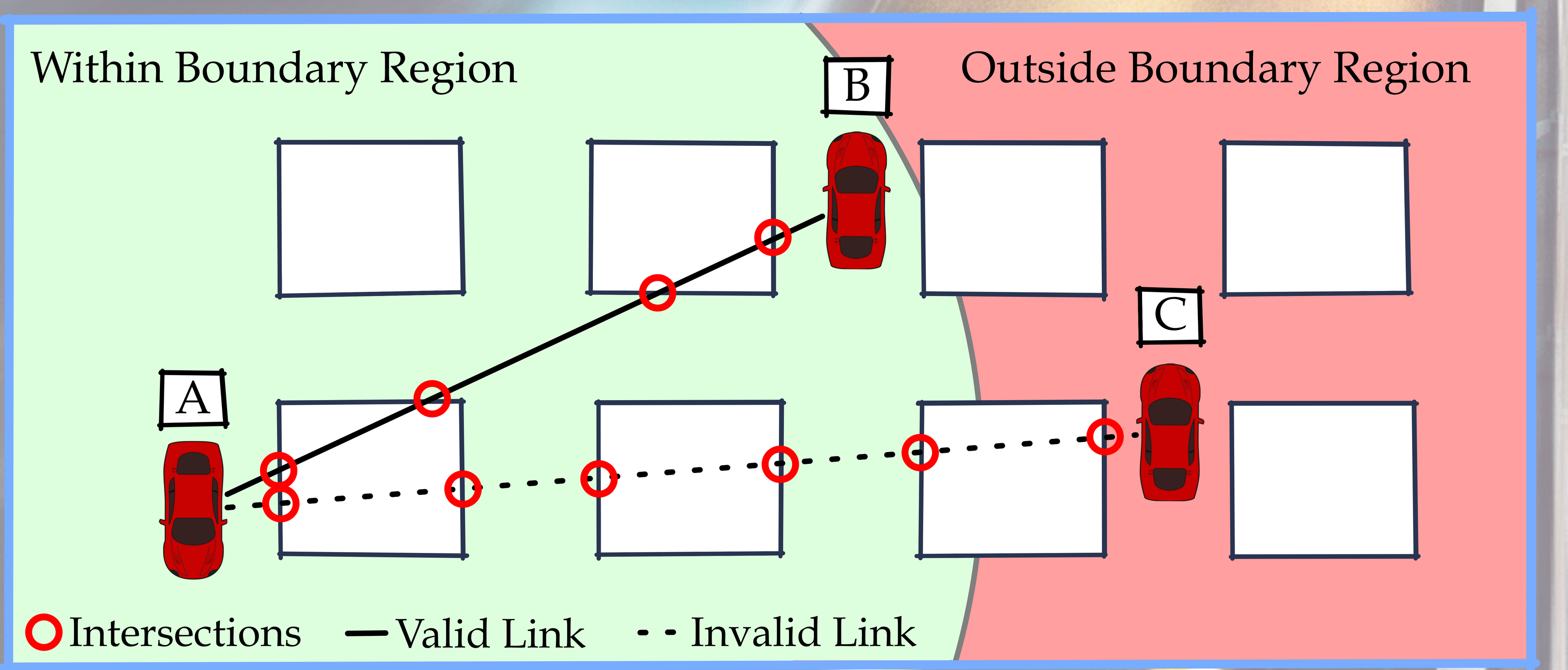

Fig. 1 Intersections with walls and boundary regions on a map.

Our code is freely available:
https://github.com/v2x-dev/multithread-inet

4. Proposed Solution

To overcome the above problems, we **redesigned the INET functions** that are responsible for the aforementioned problems.

A **multi-thread version** of *PhysicalEnvironment* class was developed:

- That enables the **parallel calculation of the power attenuation** from all the obstacles.

A **distance boundary** was introduced in the system, to integrate the **notion of the transmission range**:

- If the distance between two vehicles is greater than the boundary, the **packets are regarded as non-deliverable**.

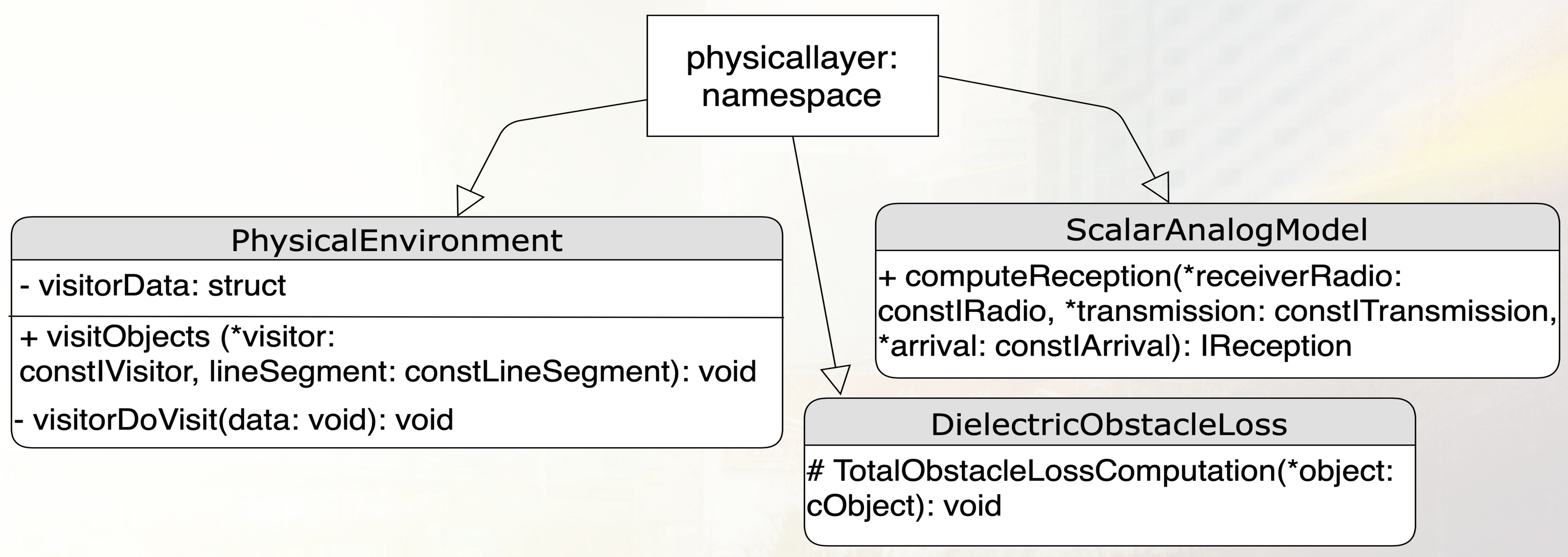

Fig. 2 The UML diagram of the modified/added functions.

5. Performance Evaluation

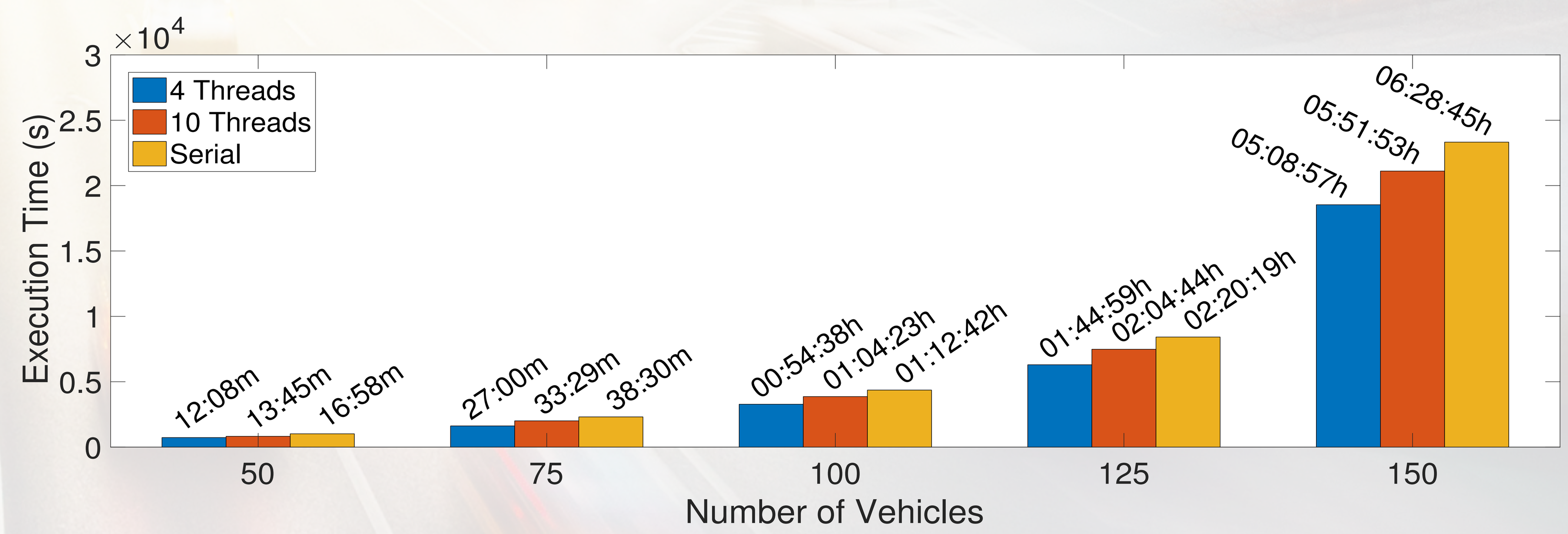

Fig. 3 The execution time, as a function of the number of vehicles.

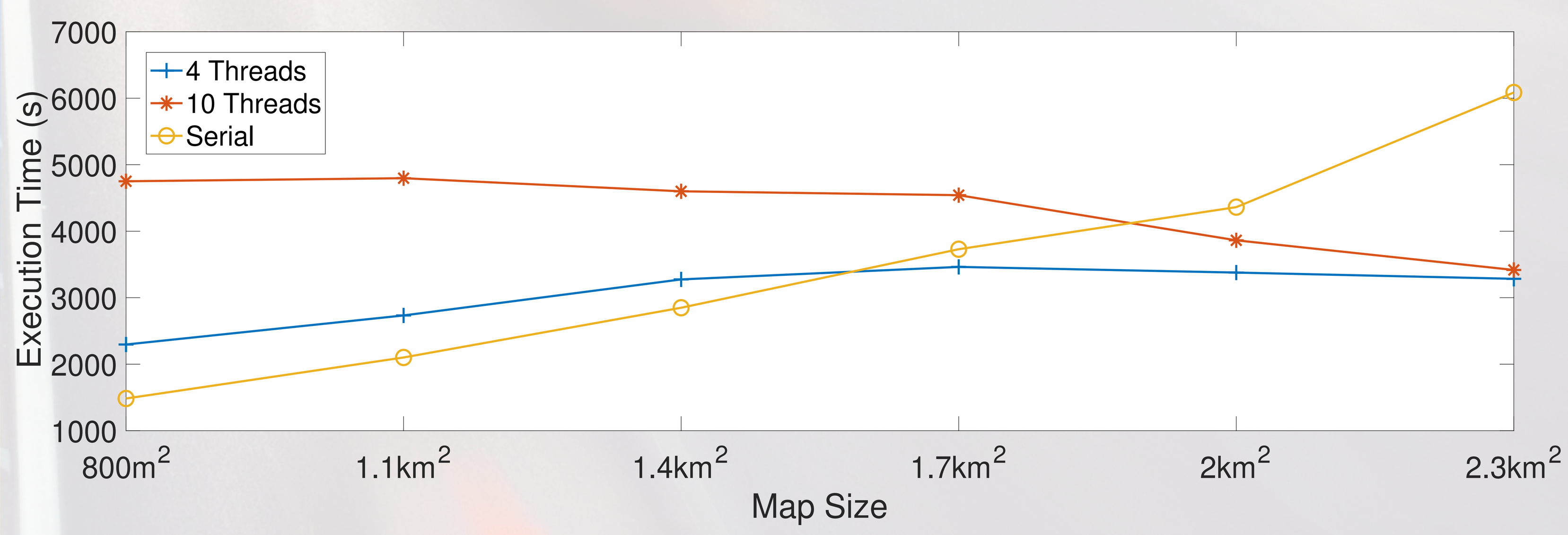

Fig. 4 The execution time, as a function of the map size.

6. Conclusions

- Several INET functions were refactored in a multi-thread fashion, to decrease the simulation time.
- The computation time was **decreased up to 43%**.
- Our implementation **ensures**:
  a. **Easy configurability** to speed up the simulation time, when required.
  b. **Seamless integration** with existing scenarios.