IVN Project Implementation Report:
5G NR and LTE-A D2D CAM-like traffic dataset

Student: Enzo Nicolás Spotorno Bieger

---

1. Introduction

The goal of the IVN project was to generate an extensive dataset to collect network metrics and statistics using LTE V2X. Early efforts focused on making Artery work with SimuLTE. However, packet transmission issues arose: only GeoNet and LTE handover packets were transmitted. This issue hindered the recording of statistics, as the simulation only reached the LTE handover logic, compromising the project's objectives.

Given SimuLTE's discontinuation and its successor, Simu5G, being actively updated, the project pivoted to Simu5G for its superior support for LTE and 5G NR networks. Although integrating Simu5G with Artery was outside the scope of the project due to dependency mismatches, Simu5G offered a modern and reliable alternative.

The simulation-related files developed for this project are available at the github repository for the project.

---

2. Selected Simulation Tools and Versions

To achieve the project objectives, the following tools and versions were used:

- **Omnet++**: 6.0.3
- **INET**: 4.4
- **SUMO**: 1.11.0
- **Simu5G**: 1.2.1
- **Veins**: 5.2

Simu5G's documentation is significantly better than SimuLTE's, and they offer a plug-and-play virtual machine (VM) with preconfigured components for easy installation and use. Local installation, however, required specific adjustments due to incompatibilities.

---

3. Local Installation Instructions

To set up a local installation, the following steps were followed:

1. **Workspace Setup**
    - Create a workspace directory: `/home/workspace`.
2. **Download Required Components**
    - Omnet++ 6.0 following official instructions.
    - SUMO 1.11.0 from SUMO documentation and add it to the system PATH.
    - Veins 5.2 extracted to `/home/workspace/simu5GWS`.
    - Simu5G 1.2.1 from GitHub extracted to `/home/workspace/simu5GWS` and renamed to "simu5G".
3. **Set Up Omnet++ Projects**
    - Open the Omnet++ IDE on the `/home/workspace/simu5GWS`.
    - Install INET 4.4 through the menu: `Help -> Install Simulation Models`.
    - Import Veins, INET, and Simu5G projects via `File -> Import -> General -> Existing Projects into Workspace`.
    - Configure project references:
        - Veins_INET → Reference INET 4.4.
        - Simu5G → Reference Veins_INET but not Veins.
    - Enable Simu5G features: `Project Properties -> Omnet++ -> Project Features` → Tick "Simu5G Cars".
    - Build all projects (`Ctrl + B`).
4. **Validation**
   Run the `cars` example in both LTE and NR folders (Simu5G/simulations) while ensuring the `veins_launchd` daemon is active.

---

4. Base Station Positioning

Accurate base station positioning is essential for realistic simulations. Options include:

1. **Real-world Data**: Often unavailable due to provider restrictions (e.g. IEEE 7835952).
2. **Standardized Grid Patterns** (e.g. JWCN Article).
3. **Cellmapper.net**: Offers detailed data but lacks an API. Manual annotation is possible but time-consuming.
4. **OpenCelliD Dataset**: Free and publicly available but less precise.

To streamline the use of real-world data, a custom script was created to convert geographic coordinates into Omnet++ coordinates, enabling realistic base station placement in simulation files.

The script can be found on the repository for the project in the “positionMapping” subdirectory, together with a README file.

---

5. Simulation Scenario

The LuSTNanoScenario from the F2MD repository was chosen for validation, leveraging SUMO for vehicle mobility. Complex scenarios are avoided due to increased computational demand without additional complexity in implementation.

Network Configuration

A custom `.ned` file and `.ini` file were created to support dynamic vehicles and base stations. Base station coordinates were derived from opencellid.org, processed through a custom script, and included in the `.ini` file.

Background Traffic

Random background cells and user equipment (UEs) were added to simulate interference, utilizing Simu5G's features. Initial simulations included VoIP (DL and UL).

D2D Multicast and Application Traffic

The current simulation also supports Device-to-Device (D2D) multicast communication, utilizing Sidelink (PC5) in network-assisted mode (mode 3 for LTE, mode 1 for 5G). In this mode, the eNodeB/gNodeB assigns resources for the multicast transmission, but the data itself is exchanged directly between devices via Sidelink. This is currently the only D2D mode supported by Simu5G.

Application Traffic: Injecting CAM-like Messages

To simulate real-world traffic scenarios, we implemented an application to inject "almost" CAM-like messages into the network. These messages resemble Cooperative Awareness Messages (CAM) commonly used in vehicle communication systems. Our application functions by:

1. **Leveraging Existing SUMO Simulations:** We utilize CSV tables created from an Artery simulation using the same SUMO scenario. These tables map vehicles with the timestamps and sizes of their CAM messages. The code that

had to be inserted into Artery to generate this CSV files is documented in the github repository for the project, in the “codeInsertedInArtery.txt” file.

2. **Traffic Injection:** Based on the information in the CSV tables, the application sends datagrams with the specified size at the corresponding timestamps within the Veins simulation. An additional receiver application verifies message arrival and logs reception and size information.
3. **Stochasticity:** A key challenge lies in controlling the inherent randomness of the two simulators (Artery and Veins) to ensure both simulations experience the same traffic patterns. Since we only care for the sthochasticity of SUMO (responsible for controlling the cars) we can control it by setting a seed in the .launchd.xml file for the simulation with the <seed value="1000" /> parameter, according to SUMO documentation.
4. **Future Work:** While this approach effectively injects CAM-like traffic, it can be further improved to include messages similar to Cooperative Perception Messages (CPM) used in Cellular V2X (C-V2X) communication. Future works include injecting all the contents of the message (not only a datagram with the same size in bytes).

---

6. Addressing Simulation Bugs

A critical bug caused SUMO co-simulation to stop after some seconds, preventing vehicle spawning. Analysis revealed this occurred during breaks in vehicle spawning. A workaround involved introducing a stationary vehicle parked for the entire simulation.

Implementation

Add a parking area in the `lust.add.xml` file:

<parkingArea id="ParkAreaA" lane="--30348#5_0" startPos="41.89" endPos="51.89" roadsideCapacity="1" angle="45" length="5"/>

Add a parked vehicle in the `.rou.xml` file:

<vehicle id="randUni1024:1" type="passenger2b" depart="0.0">
<route edges="--30348#5"/>
<stop parkingArea="ParkAreaA" duration="86400"/>
</vehicle>

This ensured continuous operation, enabling uninterrupted data collection.

---

7. Data Collection and Storage

The simulation generates .vec and .sca files, which grow significantly with increased complexity. For example, one hour of the LuSTNano scenario originally produced 30 GB of raw statistics, capturing data from base stations, cars, background UEs, and background cells. To reduce file size and focus on relevant data, the simulation was configured to record statistics only from cars and base stations, resulting in a file size reduction of over 3x, and when compressed with ZIP of over 9x.

While still large, these files remain easily navigable via the Omnet++ IDE for visualization and analysis, making it advantageous to retain the raw data for in-depth investigation.

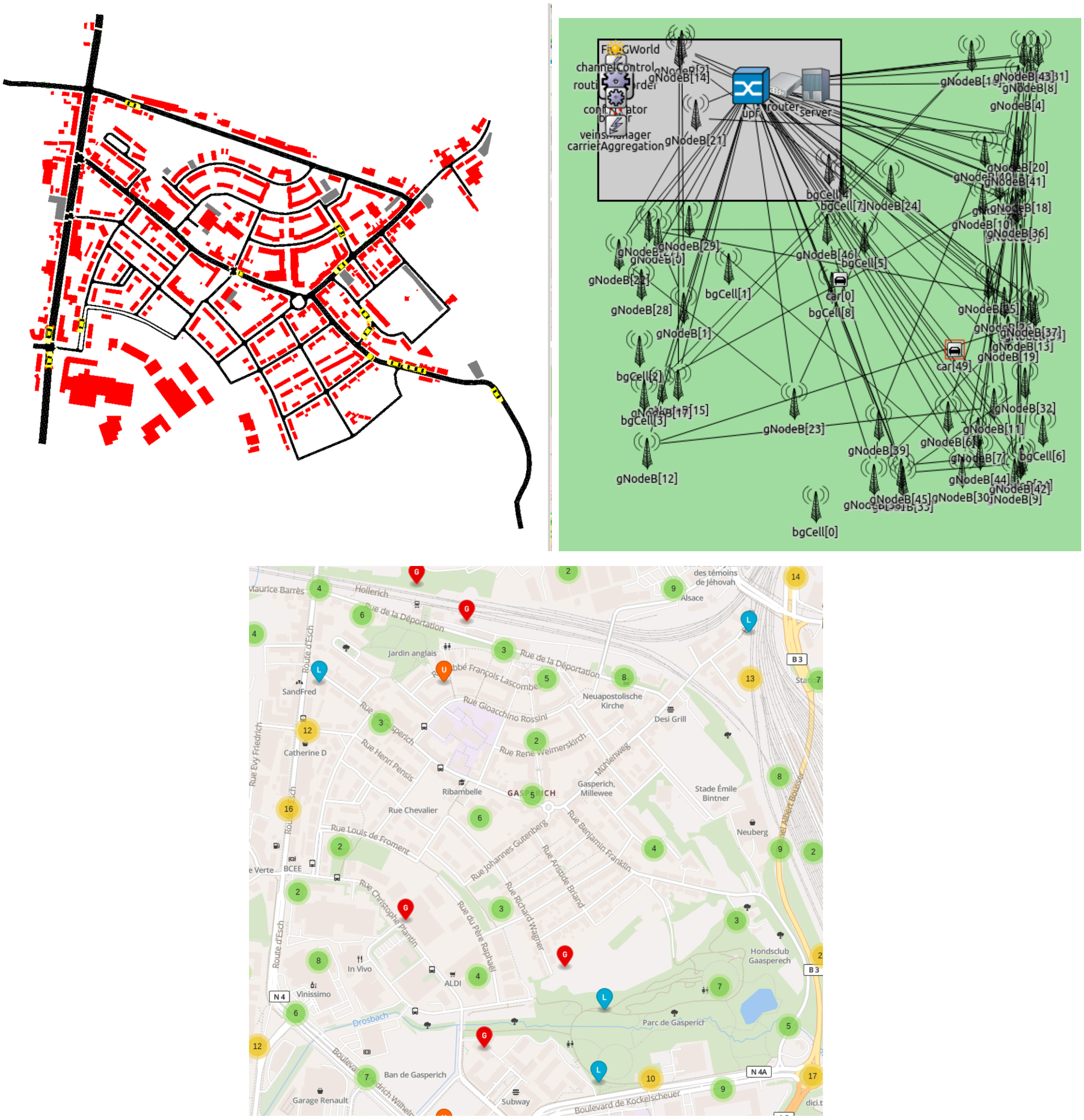
Nevertheless, a script was created to process these resulting files, clean them, organize them in a SmartData format, and send them to the IoT platform. The script is available on the github repository for the project, under the subdirectory "datasetExporting".

---

8. Conclusion

Despite initial challenges, the pivot to Simu5G and the development of tools and workarounds ensured the successful implementation of the IVN project. The project now facilitates extensive and more realistic data collection for both LTE and 5G NR scenarios, paving the way for future research.

Simulations and configurations are available on the github repository for the project, and can be transferred to local installations or the PnP VM. Nevertheless, they can also be transferred to an eventual consolidated Simu5G@Artery integration.



Top-Left figure is the SUMO LuSTNanoScenario.
Top-right figure is the Omnet++ network generated.
Bottom figure is the OpenCelliD website.

Bug Report

Title: SUMO simulation stops unexpectedly during long simulations with Simu5G and Veins

Description: When running long-duration simulations with Simu5G, Veins, SUMO, Omnet++, and INET, the SUMO simulation stops abruptly without any error message. This happens consistently in both the Simu5G PnP VM and a local installation, across various scenarios and network configurations. The Omnet++ simulation continues, but no new vehicles are spawned in SUMO, effectively halting the vehicular network simulation and preventing data collection.

Steps to Reproduce:

1. **Set up simulation:** Use the Simu5G PnP VM or follow the local installation instructions detailed in the attached project report.
2. **Scenario:** Employ any scenario with continuous traffic patterns and a longer duration (exceeding 1 hour for instance). The LuSTNanoScenario from the F2MD repository reliably reproduces this issue.
3. **Modify simulation:** Change the “cars” examples (both LTE and 5G NR) to use the lust.launchd.xml file, making sure to have all the necessary scenario files in the directory. These files can be found here. Also, remember to modify the simulation so the “sim_time_limit” is higher.
4. **Run simulation:** Initiate the simulation and observe.

Expected Result: The simulation should run continuously for the entire duration specified in the scenario, with SUMO and Omnet++ working in sync.

Actual Result: The SUMO simulation stops prematurely without any error message, while the Omnet++ simulation continues. This prevents new vehicles from being generated, hindering the simulation of the vehicular network.

Observations:

- The issue appears to occur during breaks in vehicle spawning when no vehicles are visible on the SUMO GUI.
- Introducing a stationary vehicle parked for the entire simulation in the LuSTNanoScenario (as described in the attached report) serves as a temporary workaround, but this may not be a general solution and could affect the realism of the simulation.

System Information:

- **Simu5G version:** 1.2.1
- **Veins version:** 5.2
- **SUMO version:** 1.11.0

- **Omnet++ version:** 6.0.3
- **INET version:** 4.4
- **Operating System:** Ubuntu 20.04 (PnP) or Ubuntu 22.04 (Local installation)
- **Simulation duration:** Longer (1 hour for instance)

Possible Workarounds:

- Adding a permanently parked vehicle to the simulation scenario (as detailed in the attached report). This workaround is not ideal as it may introduce artificial behavior and impact the validity of the simulation results.

Severity: High - This bug severely limits the ability to conduct realistic long-term simulations with Simu5G and Veins, preventing the collection of meaningful network traces and hindering research objectives.