

Artery V2X Simulation

Table of Contents

[Show/Hide]

- [Artery V2X Simulation](#)
- [1. Collaborative Perception in IEEE 802,11p Dataset](#)
 - [1.1. Dataset](#)
- [2. 5G NR and LTE-A D2D CAM-like Traffic Dataset](#)
 - [2.1. Selected Simulation Tools and Versions](#)
 - [2.2. Local Installation Instructions](#)
 - [2.2.1. Workspace Setup](#)
 - [2.2.2. Download Required Components](#)
 - [2.2.3. Set Up Omnet++ Projects](#)
 - [2.2.4. Validation](#)
 - [2.3. Base Station Positioning](#)
 - [2.4. Simulation Scenario](#)
 - [2.5. Addressing Simulation Bugs](#)
 - [2.6. Data Collection and Storage](#)
 - [2.7. Discussion](#)
 - [2.8. Bug Report](#)
 - [2.9. Accessing the dataset via LISHA's IoT Platform](#)
- [3. Ecologic Datasets](#)
 - [3.1. Geneva Motorway](#)
 - [3.1.1. Smartdata Model](#)
- [4. Collaborative Perception in 5G-NR D2D Dataset](#)
- [Review Log](#)

1. Collaborative Perception in IEEE 802,11p Dataset

This dataset is a recording of approximately 60 minutes of vehicles driving in the Artery simulator. Figure 1 shows the map vehicles drive in. Each vehicle follows a random driving pattern. We record vehicle dynamics. Data is collected every 100 ms. The vehicular dynamics used in this simulation are available in [AutoX - Vehicle Dynamical Model](#).

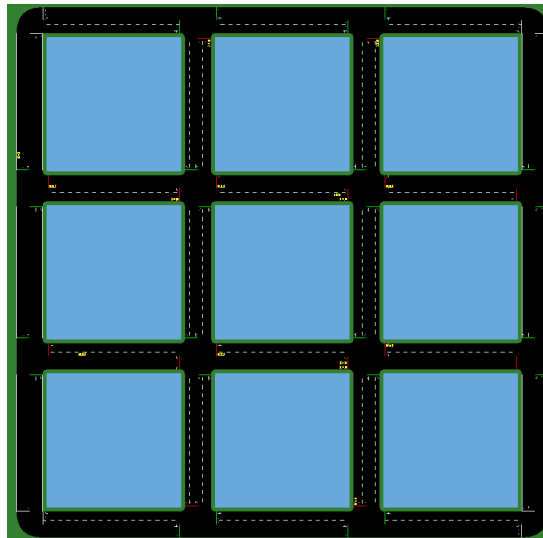


Figure 1: Simulation map.

:

The data captured in the simulation is described in the table below with the unit for SmartData and the related SI unit in accordance with: AutoX - Vehicle SmartData Model.

1.1. Dataset

The dataset can be downloaded [here](#) along with an example reader. The data is stored in the csv format with the columns timestamp, id, lat, lon, alt, x, y, z, speed, heading, yawrate and acceleration.

Note that, while the files are numbered sequentially, the vehicle IDs are not sequential. Vehicle IDs are generated based on ETSI ITS pseudonym rules. The timestamps of the data are in simulation time, following the 10Hz generation. The table below shows the vehicle IDs in order, and the start and end timestamps.

Description	Values
Vehicle IDs	1124, 1272, 1947, 680, 1426, 1503, 532, 757, 313, 2101, 1580, 1870, 1349, 2024, 1047
Start Timestamp (t0)	1748768400000000
End Timestamp (tf)	1748772002000000

2. 5G NR and LTE-A D2D CAM-like Traffic Dataset

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 [GitLab repository for the project](#).

The experiment page detailing the related problems and goals can be seen [here](#).

2.1. 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.

2.2. Local Installation Instructions

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

2.2.1. Workspace Setup

- Create a workspace directory: /home/workspace.

2.2.2. 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".

2.2.3. 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).

2.2.4. Validation

Run the cars example in both LTE and NR folders (Simu5G/simulations) while ensuring the veins_launchd daemon is active.

2.3. Base Station Positioning

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

- Real-world Data: Often unavailable due to provider restrictions (e.g. [IEEE 7835952](#)).
- Standardized Grid Patterns (e.g. [JWCN Article](#)).
- [Cellmapper.net](#): Offers detailed data but lacks an API. Manual annotation is possible but time-consuming.
- [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.

2.4. 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:

- **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.
- **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.

- **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 stochasticity 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.
- **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).

2.5. 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.

2.6. 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".

2.7. Discussion

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 GitLab 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.

2.8. Bug Report

This is the bug report sent to the Simu5G team with this [GitHub Issue](#)

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:

- Set up simulation: Use the Simu5G PnP VM or follow the local installation instructions detailed in the attached project report.
- 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.

- 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.
- 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.

2.9. Accessing the dataset via LISHA's IoT Platform

Below is the mapping of variables to units:

Variable	Unit	Comment
transmissionState:vector	0xF8000006	
passedUpPk:vector(count)	0xF8000006	
droppedPkWrongPort:vector(count)	0xF8000006	
droppedPkBadChecksum:vector(count)	0xF8000006	
servingCell:vector	0xF8000006	
txPk:vector(packetBytes)	0x03000000	# bits
rxPkOk:vector(packetBytes)	0x03000000	# bits

passedUpPk:vector(packetBytes)	0x03000000	# bits
receivedPacketFromLowerLayer:vector(packetBytes)	0x03000000	# bits
posLongCar:vector	0xC4B24924	# *pi/180
posLatCar:vector	0xC4B24924	# *pi/180
posZCar:vector	0xC4964924	
packetSent:vector(packetBytes)	0x03000000	# bits
rcvdPkFromHI:vector(packetBytes)	0x03000000	# bits
receivedPacketFromUpperLayer:vector(packetBytes)	0x03000000	# bits
sentPacketToLowerLayer:vector(packetBytes)	0x03000000	# bits
packetReceived:vector(packetBytes)	0x03000000	# bits
harqErrorRate_1st_UL:vector	0xF8000001	
harqErrorRateUl:vector	0xF8000001	
harqErrorRate_2nd_UL:vector	0xF8000001	
harqTxAttemptsUl:vector	0xF8000006	
sentPacketToUpperLayer:vector(packetBytes)	0x03000000	# bits
rlcPacketLossUl:vector	0xF8000001	
harqErrorRate_3rd_UL:vector	0xF8000001	
rlcPacketLossD2D:vector	0xF8000001	
rlcPacketLossTotal:vector	0xF8000001	
rlcPduPacketLossD2D:vector	0xF8000001	
queueLength:vector	0x03010000	# pk
queueBitLength:vector	0x03000000	# bits
distance:vector	0x84964924	
incomingPacketLengths:vector	0x03000000	# bits
queueingTime:vector	0x84925924	
outgoingPacketLengths:vector	0x03000000	# bits
avgServedBlocksUl:vector	0x03020000	# blocks
avgServedBlocksDI:vector	0x03020000	# blocks
measuredSinrUl:vector	0xF8000000	
measuredSinrDI:vector	0xF8000000	
alertDelay:vector	0x84925924	
alertRcvdMsg:vector	0x84925924	
averageCqiUl:vector	0x03030000	
rcvdSinrUl:vector	0xF8000000	
macDelayUl:vector	0x84925924	
averageCqiD2D:vector	0x03030000	

macDelayDI:vector	0x84925924	
rlcPduThroughputD2D:vector	0x03040000	# bps
rlcPduDelayD2D:vector	0x84925924	
rlcThroughputD2D:vector	0x03040000	# bps
rlcDelayD2D:vector	0x84925924	
incomingDataRate:vector	0x03040000	# bps
outgoingDataRate:vector	0x03040000	# bps

Below the transformations made to the variables before sending them to the platform:

Variable	Transformation	Comment
txPk:vector(packetBytes)	lambda x: x * 8	# bytes*8
rxPkOk:vector(packetBytes)	lambda x: x * 8	# bytes *8
passedUpPk:vector(packetBytes)	lambda x: x * 8	# bytes *8
receivedPacketFromLowerLayer:vector(packetBytes)	lambda x: x * 8	# bytes*8
posLongCar:vector	lambda x: x * 3.141592653589793 / 180	# *pi/180
posLatCar:vector	lambda x: x * 3.141592653589793 / 180	# *pi/180
packetSent:vector(packetBytes)	lambda x: x * 8	# bytes*8
rcvdPkFromHI:vector(packetBytes)	lambda x: x * 8	# bytes*8
receivedPacketFromUpperLayer:vector(packetBytes)	lambda x: x * 8	# bytes*8
sentPacketToLowerLayer:vector(packetBytes)	lambda x: x * 8	# bytes*8
packetReceived:vector(packetBytes)	lambda x: x * 8	# bytes*8
sentPacketToUpperLayer:vector(packetBytes)	lambda x: x * 8	# bytes*8
rlcPduThroughputD2D:vector	lambda x: x * 8	# Bps *8
rlcThroughputD2D:vector	lambda x: x * 8	# Bps *8

There are multiple modules that present some of these variables, the modules list can be found in the sendToloT script at the repository for the project, it is too big to be displayed here.

The data and time chosen to be the start of the timestamps were: 2024-12-20 00:00:00. The timestamps are incremented over this. Being the end time 2024-12-20 12:00:00

When the module is a car the x, y, and z parameters of the smartdata are the actual coordinates, in ECEF. If not a car, everything is set to zero. 'error' is set to zero, and 'confidence' is set to 1. 'signature' is 1 + the index of the module in the modules' list, and 'dev' is the index of the variable in the mapping of units.

The instructions to configure the environment to run the simulations and generate the dataset are also in the repository for the project

3. Ecologic Datasets

3.1. Geneva Motorway

Digital Twin - Geneva Motorway (DT-GM) is a microscopic traffic simulation dataset and scenario for SUMO that reproduces a section of the Geneva motorway in real time using fine-grained traffic counter data (minute resolution) obtained via the Swiss ODPMS platform. The repository provides all configuration files, network definitions, and control scripts needed to stream actual motorway measurements (flows, speeds, and vehicle classes) into a running SUMO simulation, enabling on-the-fly calibration of traffic volumes and dynamic rerouting during system run-time. This setup supports experiments such as SCENARIO-I, where real traffic flows are ingested continuously to create a synchronized digital twin of the physical motorway, intended for research on real-time traffic management, control optimization, and predictive analytics.

3.1.1. Smartdata Model

The Geneva Motorway dataset is available at the platform, deviot and iot. The aggregated eco-scores and speeds, can only be accessed at the dev platform for now, since the workflows are only available at the dev environment. The smartdata representation can be found below:

Name	SmartData Unit	dev	SmartData Quantity	Min	Max	Sampling Rate	Semantics
Motion Vector	0x4DFD9654 (F32)	0	-	0	1000	1Hz	The motion vector for the vehicle.
CO2	0x4EE5E4E4 (F32)	43	kg	0	0.001	1Hz	The instant CO2 consumption of the vehicle.
Fuel Consumption	0x4EE5E4E4 (F32)	42	kg	0	0.001	1Hz	The instant Fuel consumption of the vehicle.
Not Raining Average Eco-Score	0xF8000001 (I32)	0	%	0	100	event-driven	Ecologic Driving score if not raining, averaged per hour.

Name	SmartData Unit	dev	SmartData Quantity	Min	Max	Sampling Rate	Semantics
Raining Average Eco-Score	0xF8000001 (I32)	1	%	0	100	event-driven	Ecologic Driving score if raining, averaged per hour.
Not Raining Average Speed	0xC4963924 (F32)	0	m/s	0	100	event-driven	The average speed, if raining, of vehicles in meters per second, averaged per hour.
Raining Average Speed	0xC4963924 (F32)	1	m/s	0	100	event-driven	The average speed, if raining, of vehicles in meters per second, averaged per hour.

- The dataset timestamps consist of $t_0 = 1648076403000000$ and $t_1 = 1648114640000000$.
- The signatures, which correspond to vehicle ids, can be found in this file <https://lisha.ufsc.br/dl9392>
- Some signature examples are: flow_truck1_temporary_N_S.0, flow_car1_E_NS.1, caliCar1_N_ES147_0.0.0, flow_car1_temporary_N_S.2, caliCar1_N_ES147_0.0.1, flow_car1_temporary_N_S.3.

4. Collaborative Perception in 5G-NR D2D Dataset

Review Log

Ver	Date	Authors	Main Changes
1.0	Jun 2025	Mateus Lucena	Collaborative Perception in IEEE 802.11p Dataset

1.1	Jan 2026	José Luis Conradi Hoffmann	Quality review
1.2	Jan 2026	Enzo	5G NR and LTE-A D2D CAM-like Traffic Dataset
1.1	Jan 2026	Lenzi	Ecologic Dataset
1.4	Jan 2026	Edson	Collaborative Perception in 5G-NR D2D in Artery Dataset