CSE 4616 – Wireless Networks Lab

Assignment-2

Performance Evaluation of Vehicular Communications under Urban and Freeway Scenarios

Table of Contents:

🎯 Objective

The goal of this assignment is to evaluate the performance of IEEE 802.11p (DSRC) under two realistic vehicular scenarios using OMNeT++/Veins. You need to collect and analyze communication metrics to understand how traffic density and road topology affect message dissemination reliability and timeliness.

🔧 Simulation Environment Setup

Step-by-Step Instructions:

The package `veins.zip` (available on Google Classroom) contains the necessary `Sumo-related files`.

1.1. **Delete all the content of veins example scenarios and simulation scripts available at /home/src/veins/examples/veins/**

1.2. **`Extract the provided`** *`veins.zip and`* then paste the inner contents of this inside **/home/src/veins/examples/veins/** directory.

Since some source files of the `/home/src/veins` project have been modified by the course instructors to generate additional statistics required for these experiments, you must use the updated files provided. The package `veins_src_files.zip` (available on Google Classroom) contains the necessary `.cpp`, `.h`, and `.msg` files. **Replace the corresponding default files with these updated ones.**

1.3. **/home/src/veins/src/veins/modules/application/ieee80211p/DemoBaseApplLayer.h** `with provided` **DemoBaseApplLayer.h**

1.4. **/home/src/veins/src/veins/modules/application/ieee80211p/DemoBaseApplLayer.cc** `with provided` **DemoBaseApplLayer.cc**

1.5. **/home/src/veins/src/veins/modules/mac/ieee80211p/Mac1609_4.h** `with provided` **Mac1609_4.h**

1.6. **/home/src/veins/src/veins/modules/mac/ieee80211p/Mac1609_4.cc** `with` provided **Mac1609_4.cc**

1.7. **/home/src/veins/src/veins/modules/messages/DemoServiceAdvertisement.msg** `with` provided **DemoServiceAdvertisement.msg**

1.8. **/home/src/veins/src/veins/modules/messages/DemoSafetyMessage.msg** `with` provided **DemoSafetyMessage.msg**

1.9. **/home/src/veins/src/veins/modules/messages/Mac80211Pkt_m.msg** `with` provided **Mac80211Pkt_m.msg**

1.10. **/home/src/veins/src/veins/modules/messages/BaseFrame1609_4.msg** `with` provided **BaseFrame1609_4.msg**

2. Use the **lowest student ID** from your group members as the seed value in the provided **omnetpp.ini** file. (e.g., `seed-set = 210041000`)

3. Set the simulation duration to **120 seconds** by configuring the `sim-time-limit` parameter in the provided `omnetpp.ini` file. [If the generated `.csv` files exceed the maximum row limit and fail to store all recorded data, adjust the simulation time accordingly (increase or reduce as needed).]

🔧 Run the simulation

Initial Run (Required Once)

- Since some source files of the `/home/src/veins` project have been modified, the entire project must be recompiled to ensure that all dependencies are correctly built. To achieve this, you must first execute at least one simulation (any configuration) in GUI mode using the OMNeT++ IDE before running simulations from the terminal.
- **Alternative Build Method:**
  You may also build the project manually by opening a terminal in the project directory (e.g., `/home/src/veins/`) and executing:
  - `make clean`
  - `make`

Subsequent Runs in Terminal (Faster Execution):

After the `src/veins` project has been built in the IDE, you can run any configuration in terminal mode or with the IDE for all remaining experiments to achieve faster execution.

Running from the Terminal:

- Open a terminal from the project location (use Linux File Explorer, not the Omnet IDE File Explorer).
- Run the following command
  - `./run -u Cmdenv omnetpp.ini -c General`

Here,

- **`./run`**: Execute a Python executable `home/source/veins/bin/veins_run` that eventually launches the OMNeT++ simulation executable.
- **-u Cmdenv:** Runs the simulation using the command-line user interface (Cmdenv), instead of the graphical one (like Tkenv or Qtenv).
- **`-c General`**: Choose the **specific configuration** (from the `omnetpp.ini` file) to run. In this case, it's `General`.

Important Note:

- The `run` file must have executable permissions.
- If you encounter the error **"permission denied"**, assign executable permission using:
  `chmod +x run`

🚦 Traffic Rule

In road traffic systems, two main practices exist worldwide:

1. **Right-Hand Traffic (RHT) Rule**
2. **Left-Hand Traffic (LHT) Rule**

- SUMO strictly follows the **RHT rule**, where vehicles keep to the right side of the road.
- Bangladesh, like many Commonwealth countries, follows **LHT**, where vehicles keep to the left.

The table below highlights the key differences between SUMO's RHT rules and Bangladesh's LHT practice:

Traffic Rules: SUMO (RHT) vs Bangladesh (LHT)

Aspect	**SUMO (RHT)**	**Bangladesh (LHT)**
Traffic side	Keep right	Keep left
High-speed vehicles	Use the left lanes to overtake	Use the right lanes to overtake
Roundabouts	Counterclockwise	Clockwise
Driver position	Left-hand drive (LHD)	Right-hand drive (RHD)
Easier turn	Right turn	Left turn
Lane indexing in SUMO	From right to left → rightmost = 0	*(SUMO fixed as RHT, opposite of Bangladesh practice)*

📊 Performance Metrics to Analyze

Successful Message Ratio (*SMR*)

Definition:

The *Successful Message Ratio (SMR)* is the fraction of vehicles within the communication range of a sender that successfully receive its broadcast message. It is a measure of the **reliability** of message delivery in vehicular networks.

Formula:

$$SMR(\%) = (Nrecv \div Ntotal) \times 100$$

- **Where**:
  $Nrecv = Number\ of\ vehicles\ that\ successfully\ received\ the\ broadcast\ message.$

  $Ntotal = Total\ number\ of\ vehicles\ within\ the\ sender's\ communication\ range\ (potential\ receivers).$
- **Example**: If a sender has **10 neighboring vehicles** within range, and **7 of them successfully receive** the message: → SMR = 7/10 =0.7 SMR = 70%

Average Successful Message Ratio (SMR_{avg})

Definition:
The *Average SMR* is the mean reliability of broadcast message delivery across multiple transmissions or multiple senders. It represents the **average fraction of successfully received messages** by potential receivers in the network.

- $$SMRavg = \frac{\sum_{j=1}^{M} Nrecv(j)}{\sum_{j=1}^{M} Ntotal(j)}$$

Where

$M = Total\ number\ of\ broadcast\ transmissions\ (or\ sender–message\ pairs)\ considered.$

$Nrecv(j) = Number\ of\ vehicles\ that\ successfully\ received\ the\ j - th\ transmission.$

$Ntotal(j) = Total\ number\ of\ vehicles\ within\ the\ sender's\ communication\ range\ for\ the\ j - th\ transmission.$

Important Note on SMR and SMR_{avg}:

In practical wireless networks, **neither the transmitting node nor the receiving nodes can calculate the Successful Message Ratio (SMR) or the Average SMR (SMRavg) on their own** for broadcast frames (e.g., Basic Safety Messages, BSMs). This is because:

- Broadcast messages in IEEE 802.11/802.11p do not use acknowledgments (ACKs).
- The transmitter has no feedback about which neighbors successfully received the packet and which did not.
- Similarly, each receiver only knows what it successfully received, but not the total set of potential receivers or how many of them also received it.

As a result, **only an external observer with global knowledge of the network traffic** (e.g., a network administrator or monitoring system) can accurately compute SMR/SMRavg, since this requires:

1. Knowing the **total number of potential receivers** within communication range for each transmission.
2. Knowing the **actual number of nodes that successfully received** each broadcast.

In real-world deployments, this type of measurement is done through **network-wide monitoring, logging, or controlled experiments** rather than by individual nodes.

In our case, since we are working with simulations (e.g., in **Veins** with OMNeT++ and SUMO), the simulator inherently has access to all transmission and reception events across the network. Therefore, we can calculate SMR and SMRavg directly from the simulation logs, even though this would not be possible for a single node in a live system.

How to Calculate SMR_{avg} using generated .CSV:

To calculate the average Successful Message Ratio (**SMR_{avg}**) from the simulation results, use the `veinsCurrentVehiclesInfo.csv` file together with the scalar statistics (`.anf` file):

1. **Potential Receivers (from `.csv`):**
    - Veins records the required information in `veinsCurrentVehiclesInfo.csv`.
    - For each BSM transmission, identify the number of vehicles within the communication range of the transmitting node (i.e., potential receivers).
    - The total potential receivers for the entire simulation is obtained by summing all such potential receiving events across the simulation duration.

2. **Successful Receptions (from `.anf`):**
    - From the scalar statistics in the generated `.anf` file, obtain the total number of BSMs that were successfully received by all nodes during the entire simulation.

3. **Compute SMRavg:**
    - Using the two statistics above (total potential receivers and total successful receptions), calculate the average **SMR (SMRavg)** for the whole simulation time.

📊 Performance Metrics to Analyze

⏱️ No Message Interval (NoM)

Definition:
The *No Message Interval (NoM)* quantifies the **longest continuous time period** during which a receiver does not successfully receive any messages from a given sender. It highlights the worst-case gap in communication (communication outage), making it a key reliability and timeliness metric in vehicular networks.

Interpretation:

- While *Inter-Reception Time (IRT)* reports the gap between every consecutive reception, *NoM* focuses **only on the longest gap**.
- A high NoM value indicates a risk of missing critical updates for an extended period.

Example:
If a vehicle receives messages at times 1.0s, 1.1s, and 1.6s, the gaps are 0.1s and 0.5s.
Thus: NoM=0.5 s

Relevance in VANETs:

- Critical for **safety applications**, where long gaps without updates may compromise awareness.
- Useful for evaluating **worst-case performance** in high mobility or congested channel conditions.

How to Calculate *NoM* using generated .CSV:

To calculate the *No Message Interval (NoM)* from the simulation results, use the `veinsMacLayerMessageInfo.csv` and do the following.

1. **Sort the Data:**
    - Open the `veinsMacLayerMessageInfo.csv` file.
    - Sort the records first by **receiver ID**, then by **sender ID**, and finally by **reception time**.

2. **Select Sender–Receiver Pair:**
    - Choose a specific **receiver node**.
    - From the sorted data, extract the records that correspond to a particular **sender–receiver pair**.

3. **Compute Inter-Reception Gaps:**
    - For the chosen sender–receiver pair, compute the **time difference between two consecutive successful receptions** of messages.
    - If a reception is missing (i.e., a message expected but not received), this gap reflects a **missed message event**.
    - Count the number of such misses to determine the **NoM** for that sender–receiver pair.

📊 Performance Metrics to Analyze

⏳ End-to-End Delay (E2E)

- **Definition**: Time between **message generation** at the sender and **reception** at the receiver.

How to Calculate *E2E* using generated .CSV:

To calculate the **End-to-End (E2E) Delay** from the simulation results:

1. **Extract Delay Data (from `veinsApplLayerMessageInfo.csv`):**
   - The file `veinsApplLayerMessageInfo.csv` contains detailed records for each received broadcast message (e.g., BSM).
   - For every received packet, the file logs the end-to-end delay.

📡 Channel Access Delay (CAD)

- **Definition**: Waiting time before a transmitter can access the channel (due to contention/backoff).

How to Calculate *CAD* using generated .CSV:

2. **Extract Delay Data (from `veinsChannelAccessDelayInfo.csv`):**
   - The file `veinsChannelAccessDelayInfo.csv` records the channel access delay experienced by each sending node for every transmitted packet.

🔗 Metrics Insight

Metric	Indicates	Affected by
SMR_{avg} (Average Successful Message Ratio)	Reliability of broadcast message reception	- Packet collisions - Interference - Path loss/fading - Network density (number of neighbors)
NoM (No Message Interval)	Worst-case timeliness (longest gap between receptions, maximum communication outage)	- Channel congestion - Collisions - Interference - High mobility (rapid topology changes)
E2E Delay (End-to-End Delay)	Latency of message delivery (generation → reception)	- Queuing delay - Propagation delay - Channel contention - Retransmissions
CAD (Channel Access Delay)	Medium access efficiency (waiting time before transmission)	- Backoff mechanism - Contention window size - Channel load (number of competing nodes)

🔗 Relationship Between Metrics

- **SMR_{avg}** → reliability of reception.
- **NoM** → timeliness, with NoM showing worst-case gaps.
- **E2E Delay** → latency of message delivery.
- **CAD** → channel contention effects.

For **safety-critical VANETs**, desirable performance is:

- **High SMR_{avg}** (≈95%)
- **Low NoM** (timely updates)
- **Low E2E Delay** (fast delivery)
- **Low CAD** (quick channel access)

Lab Tasks

Task 1: IEEE 802.11p Safety Messaging at an Urban Intersection

Goal

Use the **junction scenario** specified in `omnetpp.ini` to analyze the **reliability and timeliness** of periodic safety messaging under **varying vehicle densities**, while keeping **default MAC/PHY parameters unchanged**.

Scenario & Configuration: (Urban Intersection Scenario)

This simulation is designed to model an urban four-way intersection under varying traffic conditions. You have to specify the road network topology, vehicle traffic profiles, and communication traffic characteristics to evaluate system performance and cooperative awareness among vehicles.

- **Road Network Topology**:

  You will use a four-way intersection, with three lanes per approach for this task. Each lane is 500 m in length. For detailed reference, consult `500m_3lanes_static_traffic_light_junction.net.xml`.

- **Road Traffic Profile:**

  The simulation will be conducted under different traffic density levels. Refer to `route_junction.rou.xml` for detailed examples on how to configure and control:

  - **Vehicle Traffic Density Parameters:** You have to control the following parameters to introduce diversity in the road traffic profile:
    - Vehicle Generation Rate
    - Starting Position
    - Lane Assignment
    - Route Selection
  - **Network Traffic Density Control**
    The intersection has **3 lanes per approach**. To simplify the calculation of instantaneous density throughout the simulation (though this is not a fully realistic traffic pattern), define traffic density **per lane** and then scale it systematically. By reducing the `period` value, traffic density increases:
    - **Example:** If each lane flow uses `period="10"`, then:
      - ≈ **0.1 vehicles/second per lane**
      - → **0.3 vehicles/second per approach** (3 lanes)
      - → **1.2 vehicles/second total** for all 4 approaches
- **Vehicle Profiles:** To introduce diversity in the road traffic profile, you may control the following parameters:
  - **Maximum speed** (set relatively lower for urban intersections, compared to highways)
  - Maximum acceleration (if necessary)
  - Maximum deceleration (if necessary)
  - Other relevant attributes (if necessary)

- **Communication Traffic Profile (Application):**
  - The simulation must ensure that each vehicle broadcasts Basic Safety Message (`BSM`) to support cooperative awareness and safety. The broadcasting periodicity should be set to **10 Hz (every 100 ms or 0.1 s).**
  - **Application Layer Setup**
    - **Application type**:
      - The demo app for WAVE message handling and SUMO interaction is `"TraCIDemo11p"`.
    - **Beaconing**:
      - You must configure the required parameter to enable periodic safety beacons (BSM). Set `beaconUserPriority` to the Highest value.
      - `*.node[*].appl.sendBeacons = true`
    - **Channel usage**:
      - You must configure the required parameter to ensure all data uses the **Control Channel (CCH)** only

Reproducibility Notes

- Use **fixed seeds** for repeatability.
  - `seed-set = <your student ID>`
- **Use the same simulation duration:**
  - `sim-time-limit = 75s`.
- Keep MAC/PHY parameters at their defaults for **all** runs. You vary **only the traffic density**.
  - **802.11p defaults (do not change for this task):**
    - *.**.nic.mac1609_4.txPower = 20mW
    - *.**.nic.mac1609_4.bitrate = 6Mbps
    - *.**.nic.phy80211p.minPowerLevel = -110dBm

Steps

1. **Select Scenario**

   Ensure the junction launch config is active in `omnetpp.ini`:
   Use **junction** config:
   ```
   *.manager.launchConfig = xmldoc("simulation_junction.launchd.xml")
   ```

2. **Define Traffic Density in SUMO**
   - Create several **traffic demand presets** for the intersection scenario, covering a wide range of densities:
   - Use SUMO's `<flow />` definitions inside the **route file (`.rou.xml`)** to specify vehicle generation for each preset.
   - Configure **per-lane flows** with appropriate vehicle rates, routes, and types so that:
     - Most approaches and lanes of the intersection are actively used.
     - Traffic density increases systematically across presets by adjusting **flow rates** (vehicles/hour) or **headways** (time between vehicles).
3. **Metrics Recording & Analysis**
   - **SMRavg (Average Successful Message Ratio)** – reliability of message delivery.
   - **NoM (No Message Interval)** – worst-case inter-reception gap.
   - **E2E Delay (End-to-End Delay)** – per-packet latency.
   - **CAD (Channel Access Delay)** – channel contention time.

Metrics Recording & Analysis Procedure

Average Successful Message Ratio (SMR_{avg})

1. **SMR vs. Density**
    - Run the simulation under **different density configurations**.
    - Compute **average SMR (SMRavg)** for each density.
    - Plot **SMRavg vs. density** to observe how reliability changes with density.

2. **Classification of SMR Results (for further analysis):**

    After computing SMRavg, classify the results into **three reliability categories**. You may adjust the thresholds yourself by examining the overall SMRavg vs. density graph and dividing it into three ranges, based on where sharp or moderate drops in reliability occur. A sample guideline is given below:

    - **Case A (High reliability):** SMRavg ≥ 85% (or 90% if applying stricter safety criteria).
    - **Case B (Moderate reliability):** SMRavg ~75–85%.
    - **Case C (Low reliability):** SMRavg below 75%.

    Note:

    - *These ranges are not absolute standards but are suggested for analysis.*
    - *Thresholds are **application dependent** – e.g., collision warnings may require ≥ 90% reliability, while less time-critical safety messages (e.g., hazard alerts) can tolerate lower values.*

3. You must keep a record of the traffic density profile and other simulation parameters used in each case (Case A, Case B, Case C).

    These settings will be explicitly referenced when analyzing NoM and E2E delay results in the subsequent tasks.

No Message Interval (NoM)

1. NoM (No Message Interval) Analysis in the Network

Step 1 – Compute NoM statistics

- For each traffic density level within each case (A, B, C), calculate:
  - **Mean NoM** (average across all sender–receiver pairs)
  - **Median NoM** (50th percentile)
  - **95th Percentile NoM** (near worst-case, excluding extreme outliers)
  - **Max NoM** (absolute worst-case interval observed)

Step 2 – Compare across densities and cases

- **Within each case (A, B, or C):** compare how NoM values change when density increases (low → medium → high).
- **Between cases (A vs. B vs. C):** compare NoM values at similar densities to analyze how different reliability levels affect both **average performance** and **worst-case timeliness**.

Step 3 – Summarize findings

- Present results using tables and plots (e.g., bar charts for Mean/Median/95th percentile, markers for Max).
- Clearly highlight how **higher density** and **lower reliability** lead to **longer NoM intervals**, indicating worse communication timeliness.

Visualization Guidance:

- Use a **grouped bar chart** (x-axis = Case A/B/C, y-axis = NoM value) with bars for Mean, Median, and 95th percentile.
- Highlight the **Max NoM** separately (e.g., marker or annotated bar) to emphasize absolute worst-case behavior.
- Add a **CDF plot of all NoM values** for each case (Case A, Case B, and Case C) to show distribution beyond summary statistics.

2. CDF plot of all NoM values in the Network

Include a **CDF plot of all NoM values** to visualize distribution beyond the averages.

- ## How to Plot the CDF of NoM Values

Step 1: Collect the raw NoM values from the simulation for each individual sender–receiver pair (not the aggregated statistics such as Mean, Median, or Max NoM at the network level).

Step 2: Sort the values in ascending order.

Step 3: For each value x, compute the cumulative probability:

$$F(x) = \frac{\#\{NoM <= x\}}{N}$$

where N is the total number of NoM samples.

Step 4: Plot the sorted NoM values on the x-axis and the cumulative probability $F(x)$ on the y-axis.

- **How to read the plot:**
    - A **steep curve** (rises quickly to 1) → most vehicles experience short NoM gaps.
    - A **long flat tail** → some vehicles experience very large gaps (worst-case reliability issue).
- **Practical Tip:**
    - In **Python**, use `numpy.sort()` and `matplotlib` to create the plot.
    - In **Excel**, sort the values, assign rank = (row number ÷ total rows), and insert a scatter plot.

- ## How to Calculate the 95th Percentile NoM and Why

Step 1: Compute NoM per sender–receiver pair

- For each sender–receiver pair, calculate the **NoM** = the longest gap between two consecutive successful receptions.
- This value represents the *worst-case interval* for that specific pair.

Step 2: Aggregate all NoM values in the network

- Collect all pairwise NoM values into one dataset.
- This produces a distribution: some pairs will have short gaps, others much longer.

Step 3: Sort the NoM values

- Arrange all NoM values in ascending order.

Step 4: Locate the 95th percentile value

- The 95th percentile is the value below which **95% of the data falls**.
- Practically, if there are N pairs:
  $Index = 0.95 \times N$
  The value at this index (in the sorted list) is the **95th percentile NoM**.

Why this is important

- The **maximum NoM** shows the absolute worst case, but it may come from a single rare outlier.
- The **95th percentile NoM** gives a “near worst-case” that is more representative of the overall network — meaning 95% of vehicles experience better performance, and only 5% are worse.
- This avoids overemphasis on extreme cases, while still capturing the **tail behavior** critical for reliability analysis.

End-to-End Delay (E2E)

Conditional E2E Delay Analysis (by SMR category)

- Using the Case A/B/C classification from SMR, analyze E2E delay results.
- For each case, compute:
  - Mean E2E Delay
  - Median E2E Delay
  - 95th Percentile E2E Delay

- Plot E2E Delay CDFs separately for Case A, Case B, and Case C.
- Compare across categories to quantify how lower reliability (SMRavg) correlates with higher end-to-end delays.

Channel Access Delay (CAD) and Its Relation to E2E Delay

Conditional CAD Analysis (by SMR category)

- Using the Case A/B/C classification from SMR, analyze Channel Access Delay (CAD).
- For each case, compute:

  - **Mean CAD**
  - **Median CAD**
  - **95th Percentile CAD**

  Visualization

  - Plot **CAD CDFs** separately for Case A, Case B, and Case C.

Correlation with E2E Delay

- Compare the **CAD trends** with the **E2E Delay results** (from the previous task).

  Provide a clear justification for your observations:

  - Explain whether increases in CAD (caused by higher channel contention) also lead to increases in E2E delay.
  - Estimate and comment on the **percentage contribution of CAD to the total E2E delay** (e.g., Mean CAD ÷ Mean E2E × 100%).
  - Specifically, verify if:
    - **Case C (Low Reliability):** Both CAD and E2E delays are significantly higher, with long delay tails.
    - **Case A (High Reliability):** Both CAD and E2E delays remain low and stable.

Saturation Indicators

The network can be considered saturated when multiple of the following conditions occur simultaneously:

- SMRavg shows a sharp drop compared to lower-density runs.
- NoM exhibits large spikes (very long reception gaps).
- E2E Delay and CAD distributions shift noticeably rightward (longer tails in the CDF).

Conclusion

- Discuss trade-offs among:
    - **SMRavg** (reliability),
    - **NoM** (worst-case timeliness),
    - **E2E Delay** and **CAD**(latency).

- Identify the **scalability limit** of IEEE 802.11p / DSRC: the approximate **maximum number of active vehicles** at which the system can still maintain:

    - SMRavg ≥ 90% or 85%,
    - Acceptable E2E delay and CAD,
    - Reasonable NoM.

Task 2: IEEE 802.11p Safety Messaging on a Freeway

Goal

Use the **freeway scenario** specified in `omnetpp.ini` to analyze **reliability and timeliness** of periodic safety messaging under **varying vehicle densities**, while keeping **default MAC/PHY parameters unchanged**.

Scenario & Configuration: (Freeway Scenario)

This simulation is designed to model a freeway with 3 lanes per direction (total 6 lanes) under varying traffic conditions. You have to specify the road network topology, vehicle traffic profiles, and communication traffic characteristics to evaluate system performance and cooperative awareness among vehicles.

- **Road Network Topology**:

  You will use a 1km long freeway with 2 approaching directions, with three lanes per approach for this task. For detailed reference, consult `1000m_3lanes_freeway.net.xml`.

- **Road Traffic Profile:**

  The simulation will be conducted under different traffic density levels. Refer to `route_freeway.rou.xml` for detailed examples on how to configure and control:

  - **Vehicle Traffic Density Parameters:** You have to control the following parameters to introduce diversity in the road traffic profile:
    - Vehicle Generation Rate
    - Lane Assignment
  - **Network Traffic Density Control**
    The freeway has **3 lanes per approach**. To simplify the calculation of instantaneous density throughout the simulation (though this is not a fully realistic traffic pattern), define traffic density **per lane** and then scale it systematically. By reducing the `period` value, traffic density increases:
- **Vehicle Profiles:** To introduce diversity in the road traffic profile, you may control the following parameters:
  - **Maximum speed** (set relatively higher for freeway, compared to urban intersections)
  - Maximum acceleration (if necessary)
  - Maximum deceleration (if necessary)
  - Other relevant attributes (if necessary)
- **Communication Traffic Profile (Application):**
  - The simulation must ensure that each vehicle broadcasts Basic Safety Message (BSM) to support cooperative awareness and safety. The broadcasting periodicity should be set to **10 Hz (every 100 ms or 0.1 s).**
  - **Application Layer Setup**
    - **Application type**:
      - The demo app for WAVE message handling and SUMO interaction is `"TraCIDemo11p"`.
    - **Beaconing**:

- ○ You must configure the required parameter to enable periodic safety beacons (BSM). Set `beaconUserPriority` to the Highest value.
  - ○ `*.node[*].appl.sendBeacons = true`
- **Channel usage**:
  - You must configure the required parameter to ensure all data uses the **Control Channel (CCH)** only

Reproducibility Notes

- Use **fixed seeds** for repeatability.
  - `seed-set = <your student ID>`
- **Use the same simulation duration:**
  - `sim-time-limit = 75s`.
- Keep MAC/PHY parameters at their defaults for **all** runs. You vary **only the traffic density**.
  - **802.11p defaults (do not change for this task):**
    - *.**.nic.mac1609_4.txPower = 20mW
    - *.**.nic.mac1609_4.bitrate = 6Mbps
    - *.**.nic.phy80211p.minPowerLevel = -110dBm

Steps

4. **Select Scenario**

   Ensure the junction launch config is active in `omnetpp.ini`:
   Use **junction** config:

   ```
   *.manager.launchConfig = xmldoc("simulation_freeway.launchd.xml")
   ```

5. **Define Traffic Density in SUMO**
   - Create several **traffic demand presets** for the intersection scenario, covering a wide range of densities:
   - Use SUMO's `<flow />` definitions inside the **route file (`.rou.xml`)** to specify vehicle generation for each preset.
   - Configure **per-lane flows** with appropriate vehicle rates, routes, and types so that:
     - Most approaches and lanes of the intersection are actively used.
     - Traffic density increases systematically across presets by adjusting **flow rates** (vehicles/hour) or **headways** (time between vehicles).
6. **Metrics Recording & Analysis**
   - **SMRavg (Average Successful Message Ratio)** – reliability of message delivery.
   - **NoM (No Message Interval)** – worst-case inter-reception gap.
   - **E2E Delay (End-to-End Delay)** – per-packet latency.
   - **CAD (Channel Access Delay)** – channel contention time.

Metrics Recording & Analysis Procedure

Average Successful Message Ratio (SMR_{avg})

4. **SMR vs. Density**
    - Run the simulation under **different density configurations**.
    - Compute **average SMR (SMRavg)** for each density.
    - Plot **SMRavg vs. density** to observe how reliability changes with density.

5. **Classification of SMR Results (for further analysis):**

    After computing SMRavg, classify the results into **three reliability categories**. You may adjust the thresholds yourself by examining the overall SMRavg vs. density graph and dividing it into three ranges, based on where sharp or moderate drops in reliability occur. A sample guideline is given below:

    - **Case A (High reliability):** SMRavg ≥ 85% (or 90% if applying stricter safety criteria).
    - **Case B (Moderate reliability):** SMRavg ~75–85%.
    - **Case C (Low reliability):** SMRavg below 75%.

    Note:

    - *These ranges are not absolute standards but are suggested for analysis.*
    - *Thresholds are **application dependent** – e.g., collision warnings may require ≥ 90% reliability, while less time-critical safety messages (e.g., hazard alerts) can tolerate lower values.*

6. You must keep record of the traffic density profile and other simulation parameters used in each case (Case A, Case B, Case C).

    These settings will be explicitly referenced when analyzing NoM and E2E delay results in the subsequent tasks.

No Message Interval (NoM)

3. NoM (No Message Interval) Analysis in the Network

Step 1 – Compute NoM statistics

- For each traffic density level within each case (A, B, C), calculate:
  - **Mean NoM** (average across all sender–receiver pairs)
  - **Median NoM** (50th percentile)
  - **95th Percentile NoM** (near worst-case, excluding extreme outliers)
  - **Max NoM** (absolute worst-case interval observed)

Step 2 – Compare across densities and cases

- **Within each case (A, B, or C):** compare how NoM values change when density increases (low → medium → high).
- **Between cases (A vs. B vs. C):** compare NoM values at similar densities to analyze how different reliability levels affect both **average performance** and **worst-case timeliness**.

Step 3 – Summarize findings

- Present results using tables and plots (e.g., bar charts for Mean/Median/95th percentile, markers for Max).
- Clearly highlight how **higher density** and **lower reliability** lead to **longer NoM intervals**, indicating worse communication timeliness.

Visualization Guidance:

- Use a **grouped bar chart** (x-axis = Case A/B/C, y-axis = NoM value) with bars for Mean, Median, and 95th percentile.
- Highlight the **Max NoM** separately (e.g., marker or annotated bar) to emphasize absolute worst-case behavior.
- Add a **CDF plot of all NoM values** for each case (Case A, Case B, and Case C) to show distribution beyond summary statistics.

4. CDF plot of all NoM values in the Network

include a **CDF plot of all NoM values** to visualize distribution beyond the averages.

- ## How to Plot the CDF of NoM Values

Step 1: Collect the raw NoM values from the simulation for each individual sender–receiver pair (not the aggregated statistics such as Mean, Median, or Max NoM at the network level).

Step 2: Sort the values in ascending order.

Step 3: For each value x, compute the cumulative probability:

$$F(x) = \frac{\#\{NoM <= x\}}{N}$$

Where N is the total number of NoM samples.

Step 4: Plot the sorted NoM values on the x-axis and the cumulative probability $F(x)$ on the y-axis.

- **How to read the plot:**
    - A **steep curve** (rises quickly to 1) → most vehicles experience short NoM gaps.
    - A **long flat tail** → some vehicles experience very large gaps (worst-case reliability issue).
- **Practical Tip:**
    - In **Python**, use `numpy.sort()` and `matplotlib` to create the plot.
    - In **Excel**, sort the values, assign rank = (row number ÷ total rows), and insert a scatter plot.

- ## How to Calculate the 95th Percentile NoM and Why

Step 1: Compute NoM per sender–receiver pair

- For each sender–receiver pair, calculate the **NoM** = the longest gap between two consecutive successful receptions.
- This value represents the *worst-case interval* for that specific pair.

Step 2: Aggregate all NoM values in the network

- Collect all pairwise NoM values into one dataset.
- This produces a distribution: some pairs will have short gaps, others much longer.

Step 3: Sort the NoM values

- Arrange all NoM values in ascending order.

Step 4: Locate the 95th percentile value

- The 95th percentile is the value below which **95% of the data falls**.
- Practically, if there are N pairs:
  $Index = 0.95 \times N$
  The value at this index (in the sorted list) is the **95th percentile NoM**.

Why this is important

- The **maximum NoM** shows the absolute worst case, but it may come from a single rare outlier.
- The **95th percentile NoM** gives a “near worst-case” that is more representative of the overall network — meaning 95% of vehicles experience better performance, and only 5% are worse.
- This avoids overemphasis on extreme cases, while still capturing the **tail behavior** critical for reliability analysis.

End-to-End Delay (E2E)

- **Conditional E2E Delay Analysis (by SMR category)**
  - Using the Case A/B/C classification from SMR, analyze E2E delay results.
  - For each case, compute:
    - Mean E2E Delay
    - Median E2E Delay
    - 95th Percentile E2E Delay

  - Plot E2E Delay CDFs separately for Case A, Case B, and Case C.
  - Compare across categories to quantify how lower reliability (SMRavg) correlates with higher end-to-end delays.

Channel Access Delay (CAD) and Its Relation to E2E Delay

Conditional CAD Analysis (by SMR category)

- Using the Case A/B/C classification from SMR, analyze Channel Access Delay (CAD).
- For each case, compute:

  - **Mean CAD**
  - **Median CAD**
  - **95th Percentile CAD**

  Visualization

  - Plot **CAD CDFs** separately for Case A, Case B, and Case C.

Correlation with E2E Delay

- Compare the **CAD trends** with the **E2E Delay results** (from the previous task).

  Provide a clear justification for your observations:

  - Explain whether increases in CAD (caused by higher channel contention) also lead to increases in E2E delay.
  - Estimate and comment on the **percentage contribution of CAD to the total E2E delay** (e.g., Mean CAD ÷ Mean E2E × 100%).
  - Specifically, verify if:
    - **Case C (Low Reliability):** both CAD and E2E delays are significantly higher, with long delay tails.
    - **Case A (High Reliability):** both CAD and E2E delays remain low and stable.

Saturation Indicators

The network can be considered saturated when multiple of the following conditions occur simultaneously:

- SMRavg shows a sharp drop compared to lower-density runs.
- NoM exhibits large spikes (very long reception gaps).
- E2E Delay and CAD distributions shift noticeably rightward (longer tails in the CDF).

Conclusion

- Discuss trade-offs among:
    - **SMRavg** (reliability),
    - **NoM** (worst-case timeliness),
    - **E2E Delay** and **CAD**(latency).
- Identify the **scalability limit** of IEEE 802.11p / DSRC: the approximate **maximum number of active vehicles** at which the system can still maintain:
    - SMRavg ≥ 90% or 85%,
    - Acceptable E2E delay and CAD,
    - Reasonable NoM.

Task 3: Comparative Analysis of Urban Intersection vs. Freeway Scenarios

Goal

Compare the performance of IEEE 802.11p safety messaging between an **urban intersection** and a **freeway scenario**, under varying vehicle densities. The objective is to evaluate how different mobility patterns and environments influence reliability (SMR), timeliness (NoM, E2E Delay), and channel access efficiency (CAD), while keeping MAC/PHY parameters unchanged.

Steps

After completing Task 1 (Urban Intersection) and Task 2 (Freeway), perform a comparative analysis to highlight the differences and similarities between the two environments.

Instructions

For each performance metric (**SMR, NoM, E2E Delay, CAD**), carry out the following steps:

1. **Statistical Comparison**
    - Compare **Mean, Median, 95th Percentile, and Maximum** values between Urban and Freeway results.
    - Highlight differences in both average behavior and worst-case outcomes.

2. **Distribution Comparison**
    - Plot and compare the **CDF curves** of NoM, E2E Delay, and CAD for both scenarios.
    - Identify whether one environment shows heavier "tails" (larger delays or gaps for some vehicles).

3. **Density Impact**
    - Discuss how performance changes as traffic density increases (low → medium → high).
    - Contrast whether Urban or Freeway scenarios degrade faster as density rises.

4. **Saturation Indicators**
    - Determine which scenario reaches **network saturation** earlier by checking:
        - Sudden drops in SMR.
        - Significant spikes in NoM.
        - Rightward shifts (longer tails) in E2E and CAD distributions.

Discussion Points

- Which scenario achieves **higher SMR** at similar densities?
- How do mobility patterns (stop–go at intersections vs. continuous flow on freeways) affect **NoM and E2E Delay**?
- In which scenario does **CAD contribute more strongly** to E2E Delay?
- Which scenario imposes the **stricter scalability limit** for IEEE 802.11p (maximum density before reliability or delay becomes unacceptable)?

✍️ Report Submission Guidelines

1. Submission Requirement:

Your submission must include the following components:

a) Simulation Files

- All configuration and network design files (e.g., `.ini`, `.ned`, `.anf`).

b) Statistical Data Files

- Source files used for generating graphs (e.g., Excel spreadsheets, Python scripts).

c) Report (.docx preferred)

- A well-structured document providing a logical, step-by-step analysis of observed trends for all tasks.
- Properly labeled plots with:
    - Clear titles
    - Axis labels
    - Legends
- A detailed explanation of:
    - Root causes of performance degradations
    - Other contributing factors

d) File Naming Convention

- The core folder must be zipped and named in the following format:
    - `Assignment-2_studentID1_studentID2_studentID3.zip`
- Use **ascending order of student IDs**, separated by underscores.
    - Example:
      `Assignment-2_210041101_210041112_210041122.zip`

2. Group Formation:

- Each group may have up to 3 students, but smaller groups are encouraged and will be given additional credit.
- Only one member of the group should submit the final `Assignment Content` through Google Classroom.

3. Submission Deadline:

- The assignment must be submitted on Google Classroom by [10 October 2025, 9.00 pm].