ENTS749C (Spring 2015): Vehicular Networks
Project (100 points)

This project's solution must be your own work. *Please note that if plagiarism is detected in your solutions, all the students involved in it will be subject to the appropriate university regulations*.

Carefully read the project details below. Your demo is due in class on May 12 (Tuesday); zipped file of your code and the PDF of project report are due in CANVAS, 5pm, May 15 (Friday). The report format and demo requirements are given in page 7 of this document. Late submission policy applies only for the code/report and is: up to 1-day late: 25% reduction; 1-to-2 days late: 50% reduction; more than 2 days late: 100% reduction (no points). Demo after May 12 is not allowed; if you miss the demo on May 12, you will receive no points for it.

Project Points Distribution

- Working code (with embedded clear comments): 50
- In-class demo: 10
- Project report: 40

Steps for Project Preparation and Completion

1. You must first identify a project partner and form a group of 2. One person from each group must email me theirs and their project partner's names.
2. It is preferred that this project be implemented in Matlab. However, you can use Python (SciPy modules) or other high-level programming languages, such as C and C++, if you prefer these languages instead of Matlab. Note that Matlab toolboxes and Simulink are not needed for this project.
3. Carefully read the project description below (pages 1-6), implement the required simulation code, and generate the required simulation results.
4. Demo in front of the class on May 12.
5. Complete and submit your project report and code to CANVAS before 5pm, May 15.

Project Overview

Vehicular Ad hoc NETwork (VANET) is based on V2V communications between vehicles and V2I communications between vehicles and road-side infrastructure. Vehicular mobility is a major challenge in VANET design. This project provides you with basic insights into vehicular mobility and how it impacts VANET network density and V2V/V2I connectivity.

As a future VANET designer, you must be able to implement vehicular mobility to analyze, for example, connectivity under certain road traffic conditions in certain geographic areas; routing protocols for multi-hop based applications; or, security and privacy threats and mitigations. You would need tools that ease construction and visualization of VANET scenarios for connectivity and network performance assessment. This project builds your skills towards such design needs.

This project has two parts. For the first part, you will simulate one type of vehicular mobility—freeway mobility—to assess network densities available for V2V connectivity under different road traffic volumes. For the second part, you will simulate an elemental graphical tool that visualizes and assesses two types of user-specified VANET scenarios: (*i*) V2V communications at a four-way street intersection; (*ii*) V2I communications between a vehicle and deployed RSUs.

Part 1: Simulation of VANET Mobility and V2V Connectivity

1.1. Vehicular Mobility Models

In this part of the project, we use the freeway mobility model to capture spatial restriction, spatial dependence, and temporal dependence aspects of vehicular mobility in freeways. [1]. In addition, car following behavior model and lane changing behavior model are needed to capture how a vehicle moves behind another vehicle in a lane or moves to an adjacent lane to overtake the front vehicle [2], [3]. Details of these three models are given below.

Freeway mobility model: Under this model: (*i*) each vehicle is spatially restricted to its own lane on the freeway; (*ii*) the speed of a vehicle is temporally dependent on its previous velocity; and, (*iii*) if two vehicles are on the same lane and within the safety distance (D_s), the vehicle on the behind, called the following vehicle, is spatially dependent on the vehicle ahead, called the leading vehicle. In particular, following vehicle's speed cannot exceed leading vehicle's speed.

Hence, for a vehicle *i* under freeway mobility model the intra-vehicle relation is: new_speed_i = $current\_speed_i$ + $random()*acceleration_i$, where $random()$ returns a value uniformly distributed in [-1, 1] and makes vehicle deceleration or acceleration equally probable. Furthermore, the resulting inter-vehicle relationship is: if vehicle *j* is ahead of vehicle *i* in its lane and $inter\_vehicle\_distance_{ij} \le D_s$, then $speed_i \le speed_j$.

Car following model: This model determines the speed of the following vehicle needed to separate the following vehicle from the leading vehicle by the safety distance. In particular, if a leading vehicle *j* is in front of a following vehicle *i* in a lane, then *i* will slow down to the speed necessary to maintain the safety distance D_s with *j*. This speed, $speed_i$, is equal to the minimum of $speed_j$ and $\left(-\beta + \sqrt{\beta^2 + (4\gamma D_s)}\right)/2\gamma$, where β =0.75 is the reaction time and γ= 0.0070104 s^2/m is the reciprocal of twice the maximum average deceleration of the following vehicle [2].

Lane changing behavior: In this model, the following vehicle can change lanes and overtake the leading vehicle, but only after waiting a *tolerance time* and if there is space available in an immediately adjacent lane, i.e., there is no vehicle within the safety distance D_s of the new position in the chosen adjacent lane.

1.2. Setup for Your Simulation

You must implement the above models under the simulation settings described below. Table 1 summarizes these simulation settings.

Parameter	**Value**
Freeway map	4 lanes, 5 kilometers length, one direction, 3m lane separation, 3 exit ramps, 3 entry ramps
Safety distance	D_s=10 meters (m)
Tolerance time	0 seconds (sec)
Vehicle speed	s_{min}=50 miles/hour, s_{max}=70 miles/hour
Vehicle acceleration	a_{min}=0 m/sec^2, a_{max}=±5 m/sec^2
Road traffic volume	3000 vehicles/hour/lane
Vehicle arrival/departure rate	0.833
Road traffic density	100-500 vehicles/lane

Table 1: Summary of the simulation settings for Part 1 of the project.

Figure 1 below illustrates the freeway map to be simulated. The freeway map must contain 4 lanes, each 5000m in length, and 3 entry ramps and 3 exit ramps which are commonly used by vehicles in all 4 lanes. You can also assume vehicles move in freeway lanes only in one-direction. Note that this 4-lane one-direction setting for a freeway represents a common scenario encountered in North America, Europe, and the rest of the world. For programmatic and computation simplicity, you can assume each lane is a straight line and the horizontal separation between adjacent lanes is 3m (10 feet).[1]

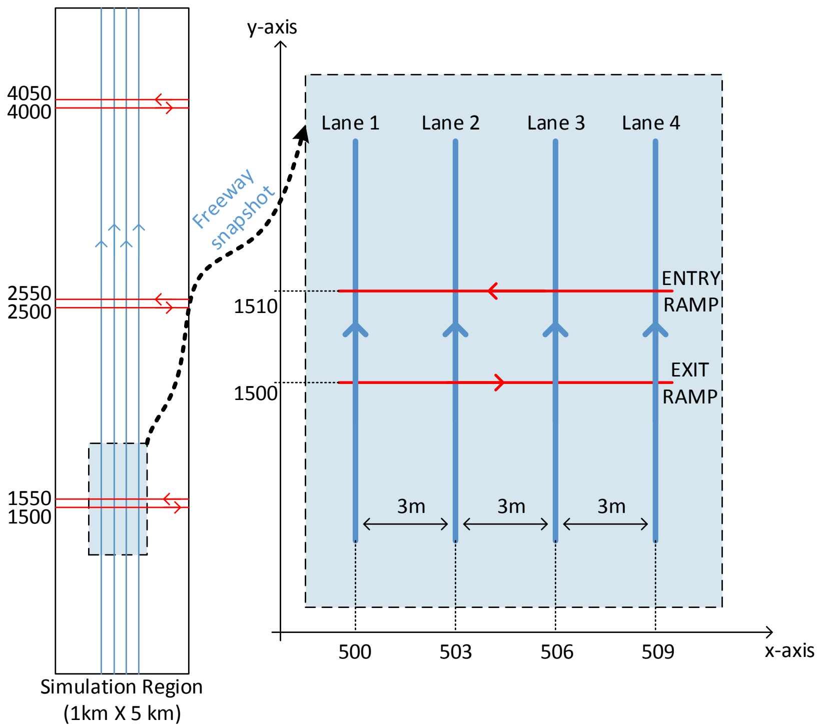


Figure 1. Illustration of 4-lane, 5km long, 3m inter-lane separation, and one-direction freeway map to be simulated.

Vehicular road traffic density per freeway lane, i.e., the average number of vehicles per lane, can vary in the range [100, 500] vehicles. When you initiate simulation you must ensure the initial positions of vehicles are *uniformly distributed* within each lane. In addition, the initial speed and acceleration of vehicles must also be assigned using a uniform distribution. For each lane, the vehicle inter-arrival and inter-departure process is kept as exponential with the vehicle arrival/departure rate of: 3000/(60×60)=0.833, assuming a freeway traffic volume of 3000 vehicles/hours/lane. Note that traffic density and volume are based on real-world traffic data.

Note: In Matlab, you can determine the arrival of new vehicles (also referred to as nodes in the rest of the document) and departure of existing nodes per lane based on exponential distribution, using expressions: `rand < arrival_rate` and `rand < departure_rate`. These expressions result in a value of *0* or *1* according to a random process and when used in your simulation: *0* means no new node arrives via a randomly chosen entry ramp (and no existing node departs freeway via a randomly chosen exit ramp); *1* means a new node arrives via a randomly chosen entry ramp (and existing node departs freeway via a randomly chosen exit ramp), respectively.

[1] This is equivalent here to the typical lane width of 10-15 feet used in practice on US freeways (dot.gov)

The network data traffic emerges from a safety application where vehicles broadcast safety messages using V2V every 100 milliseconds and with a transmission range of 50 meters. In addition, a non-safety application is also used where a vehicle must participate in a group of nodes connected over V2V for a period of time in the order of seconds. Note that two simulation settings emerging from these network applications are: (*i*) the radio communication range for a vehicle is 50 m, and (*ii*) simulation iteration step must be at a minimum of 100ms.

1.3. What Must Your Code Do?

Your code must be able to update the number of nodes in the VANET at the specified arrival/departure rate per lane; i.e., add any new nodes if they arrive at an entry ramp and remove any nodes if they depart at an exit ramp, per second. Note that the entry ramp, exit ramp, and the nodes to depart must be chosen at random in your simulation. In addition, the randomly selected departing node must be the node located at the location of the randomly selected exit ramp.

Further, your code must update node properties, including position, speed, acceleration, and lane, per iteration. Furthermore, your code must implement freeway mobility, car following, and lane changing behaviors for each of the nodes, per iteration. Moreover, your code must account for the border effect, due to the bounded 1km×5km simulation region, on vehicle mobility; this can be done by making a vehicle that reaches the border to randomly reappear in simulation region.

Your simulation code must be able to find answers for the following questions. Note that each data point in your simulation results must be an average of at least 5 iterations; each of these iterations executing for at least 10 minutes of vehicular mobility at the 100ms granularity.

1.4. What Questions Must Your Simulations Answer?

(1) What is the average V2V connectivity for a target node in this network? To begin you can choose any one node as your target node, but you must use this node throughout your simulation iterations. Run simulations and find the number of nodes within communication range of this target node and compute the average. You must do this for a road traffic density value range [100, 500] in increments of 50. Plot V2V connectivity vs. traffic density.
(2) What is the average duration the target node maintains the same 3 communication neighbors? Plot the output vs. traffic density values as in (1).
(3) What is the average number of same communication neighbors the target have for a continuous period of time during its mobility. You can choose a suitable value for this time period in the range [10, 60]. Plot the average vs. traffic density values as in (1) and (2).
(4) By increasing the V2V communication range from 50m to 100 m, what is the impact on the outcomes of (1)-(3)? Plot this separately.

For (1)-(4) note that once you choose a node to be the target node you must ensure that you do not remove it from the simulation, i.e., it should be excluded from the departing node selection.

<u>Note:</u> In the above simulations we are assuming free flow movement of vehicles. This means no traffic congestion is considered. Further, for simplicity, we are ignoring traffic lights, vehicle length, V2V communication traffic models, and V2V network protocols. These introduce programming and computational complexity beyond the course's scope. But note that enhanced VANET simulations can be done using open-source traffic simulators, such as SUMO, integrated with network simulators, such as ns-2/3 and OMNET++, or using comprehensive vehicular network simulators such as VIENS. But most of these need C++ programming language skills.

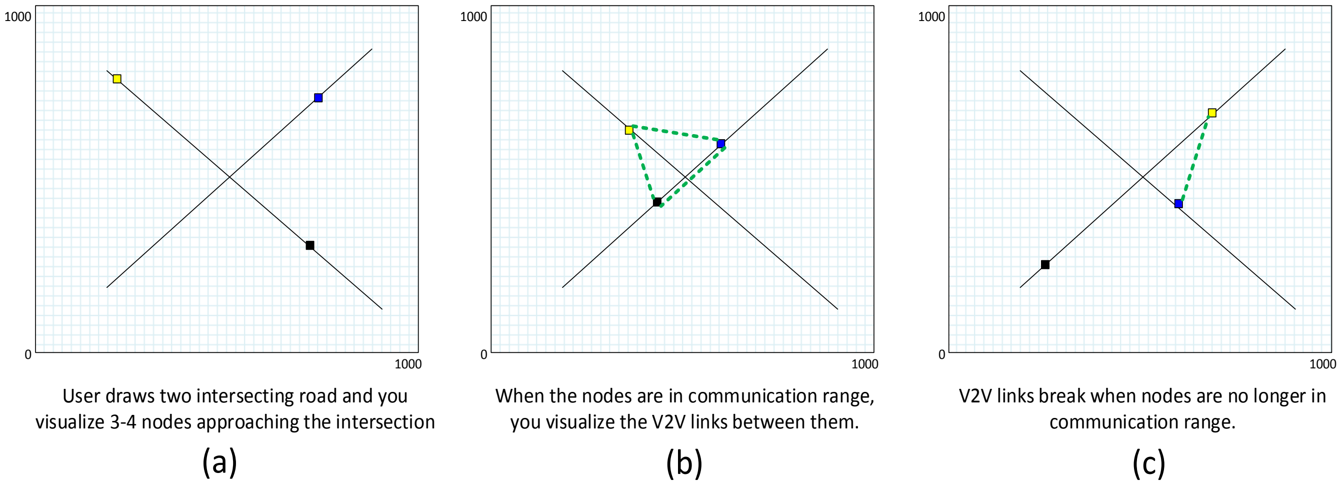


Figure 2. Illustration of the (Part 2) simulator generated graphical window intended for getting the user input and visualizing the resulting output. (a) User draws two intersecting roads and 3 vehicles are simulated on these roads, approaching the intersection at randomly selected speeds. (b) V2V links appear between nodes that are within radio communication range as they continue to move. (c) Some V2V links break when nodes go out of communication range as their movement continues towards the end of the roads.

Part 2: Simulation of VANET Visualization and Assessment Tool

In this part of the simulation, you will create a high-level simulator to visualize the V2V and V2I connectivity in streets and freeways. The simulator output is elemental and not intended for advanced networking and wireless analysis. You are, however, given flexibility to be creative and develop this simulation tool with the goal of enabling a user to understand V2V and V2I connectivity, applications, and challenges.

There are two design sub-parts here: (1) Visualizing and assessing V2V connectivity for a user-provided 4-way road intersection. (2) Visualizing and assessing V2I coverage issues for a user-provided roadway and RSU deployment. Note that in Matlab you can use `ginput` function for both of these sub-parts.

2.1. Simulating 4-Way Road Intersection and V2V Connectivity of Approaching Vehicles

When the simulator is run, it should result in a graphical window that:

(1) Allows a user to draw two intersecting roads as two intersecting straight lines.
(2) Automatically visualize 2 or 3 vehicles as squares moving at speeds—randomly selected from the range [10, 20] m/sec—on these roads towards the road intersection
(3) Visualize the V2V links between vehicles during their movement. You can assume a V2V link results if any two vehicles are within a communication range of 100m.

Figure 2 shows an illustration of the desired simulator user-interface/output window. Note that the graphical window must be large enough to simulate a 1000m × 1000m region.

2.2. Simulating V2I Coverage for a Vehicle in a Road Map with a RSU Deployment

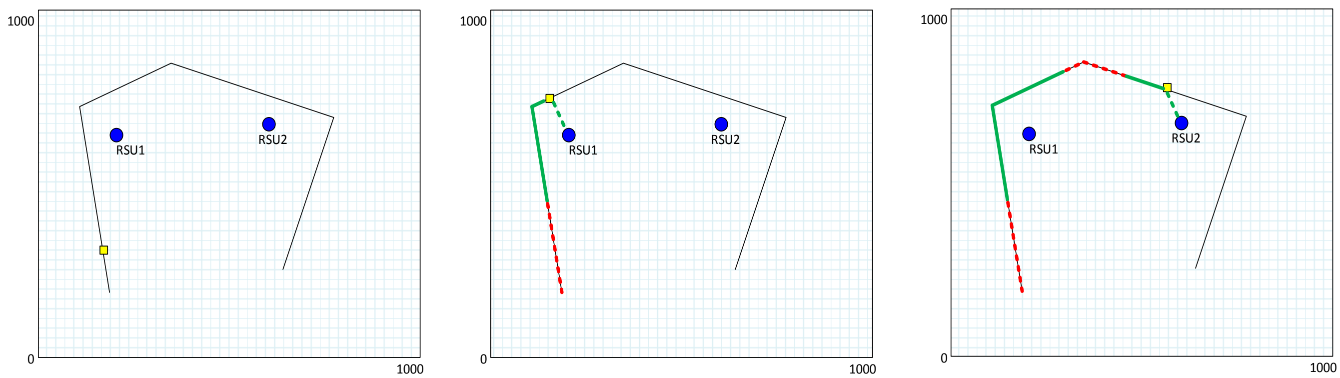
When the simulator is run, it should result in a graphical window that:

(1) Allows a user to draw a continuous road layout, i.e., a collection of interconnected straight lines, on a graphical window.
(2) Allows the user to place between 1-3 RSUs as circles around the drawn road layout.

(3) Visualize a vehicle (i.e., a square) moving at a speed—randomly selected from the range [10, 20] m/sec—from one end to the other end of the user-entered road layout.

(4) Visualize the V2I link between vehicle and the RSU with its communication range as well as visualize the V2I coverage provided by the user-placed RSUs during vehicle movement. You can assume a V2I link is established when a vehicle is within 100m range of a RSU.

Figure 3 shows an illustration of the desired simulator user-interface/output window. Note that the graphical window must be large enough to simulate a 1000m × 1000m region.



User draws a road map layout and places 1-3 RSUs in this map to assess V2I connectivity. A vehicle is automatically sent on this layout.

(a)

Vehicle moves on the layout while highlighting the V2I radio coverage. Red indicates no V2I radio coverage, green indicates coverage.

(b)

Vehicle moves towards other end of the layout communicating with all RSUs in the layout; V2I coverage concerns for RSU handoff is shown.

(c)

Figure 3. (a) User draws a continuous road layout and places RSUs; then a vehicle is simulated to move on this layout. (b) V2I links appear between vehicle and the RSU that is in radio communication range; the layout is marked with coverage (green lines on roadway) and no coverage (red, dashed lines on roadway) sectors as shown. (c) Vehicle continues to go to the other end of the road layout and the coverage map becomes useful to understand concerns such as successful handoff between RSUs.

References

[1] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks," in Proceedings of the IEEE INFOCOM, March 2003, pp. 825–835.

[2] R. W. Rothery, "Car following models," in In *N.H. Gartner, C. Messer, and A.K. Rathi, editors, Traffic Flow Theory, Chapter 4*, 2002.

[3] D. R. Choffnes and F. E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks," in Proceedings of the ACM Workshop on Vehicular Ad hoc Networks (VANET), September 2005, pp. 69–78.

Project Report Format

Each group must submit one single report in PDF format, along with your code as a zipped file, via CANVAS, before 5pm, Friday, May 15. The report must contain the following 5 sections:

(1) Abstract (less than 150 words): A paragraph stating whether you were successful, or not, in implementing the project simulations, what are your major findings, insights, results, and conclusions from the project investigation, and what language/platform did you use in your project implementation.

(2) Data Plots for VANET Mobility (use any number of pages as needed): In this section, you must include all the plots needed to answer the questions in Section 1.4. But make sure to include a caption for each of the plot, explaining what the plot is showing. Also, make sure to label the x-axis, y-axis, and provide a legend in each plot.

(3) Observations on VANET Mobility (at most 1 page): In this section, you must technically discuss any insights, findings, and conclusions you made after working through the simulation problems in Part 1.

(4) VANET Visualization and Assessment Tool Output (use any number of pages as needed): In this section, you must include your simulator tool output snapshots achieving the desired capabilities mentioned in Section 2.1 and 2.2.

(5) Instructions for Running the Code (at most 3 pages): In this section, provide user-instructions for how to run your code for Part 1 and how to run your code for Part 2.

Project Demo

Each group must demo their project results in the class period, Tuesday, May 12. You must use your laptop. Your entire demo must be <u>under 10 minutes</u>. Each demo must contain two parts:

(1) Presentation of no more than 4 PowerPoint slides, showing the plots you have for answering questions (1)-(4) in Section 1.4; also include text/plots for any interesting findings/insights/conclusions you made when you were working on Part 1 of the project.

(2) After your PowerPoint presentation, you must show the working of your simulator tool developed as per Sections 2.1 and 2.2.