VERGILIUS: A Scenario Generator for VANET

Eugenio Giordano[1,2], Enzo De Sena[3], Giovanni Pau[1], and Mario Gerla[1]
[1]University of California at Los Angeles, California, USA.
[2]DEIS, WiLab - University of Bologna, Bologna, Italy.
[3] King's College London, CDSPR, London UK *

ABSTRACT

Vehicular networks are on the fast track to become a reality either through a car manufacturer that introduces a communication device in the car electronics or through an aftermarket vendor such a GPS navigator or a in-vehicle entertainment system.

This paper introduces VERGILIUS a nouvelle urban mobility and propagation toolbox designed to streamline the mobility trace generation and path loss computation in vehicular network studies. The aim of VERGILIUS is to enable a whole new level of simulation through the introduction of Urban Maps, finely tunable motion patterns, and detailed trace analysis.

1. INTRODUCTION

Vehicular networks are clearly emerging and they will become a reality in the next few years either through after market devices such GPS navigators or a car manufacturer that integrates 802.11p or similar technology in the vehicle electronic heart.

A key factor for vehicular networks to succeed is the development of applications and protocols that fit the user needs. Studying the characteristics of the network is fundamental to understand the constraints and feasibility of protocols and services that can be run on, and even exploit, the networks formed. For example, are these networks frequently and dynamically partitioning? What is the role of the urban propagation? What are the key factors that impact applications and protocols performance? Being able to study a multitude of detailed and realistic models is essential to explore the design space and develop systems that are able to cope with a large number of very different scenarios in terms of network density, network load, and urban environment.

We here argue for a streamlined ability to systemically study urban vehicular scenarios and we propose VERGILIUS[1] a urban mobility toolbox for vehicular networks. VERGILIUS, features highly configurable mobility parameters and a number of known and new scenario metrics to enable an in depth systematic study of the causal relationship between the different scenario parameters and network performances [1].

VERGILIUS brings two important contributions to vehicular communications in urban environments. The first contribution is the macroscopic-level "tunable" vehicle motion pattern generator. It is well known that the urban traffic can be considered as the superposition of several elementary components, say, commuting pattern (to and from work), "shopping mall" pattern (say, from one mall to another in a random pattern, with pause time at each), lunch hour restaurant pattern (converging to a few popular malls), etc. The VERGILIUS motion pattern generator module can mimic any combination of traffic patterns, say commuting and "shopping mall" traffic, etc. More precisely, by changing a few parameters (e.g. number of paths used) we can increase traffic on local streets in a controlled manner. This tunable generator enables us to evaluate performance trends of various protocols (e.g. routing) under controlled traffic conditions. The output of the motion pattern generator is suitable to run microscopic-level traffic simulators such as SUMO[2] and CORSIM[3] thus allowing a effortless reproduction of a large number of mobility traces for vehicular networks. VERGILIUS automates the process of road map extraction and mobility trace creation, in contrast to todays best practices in vehicular network studies that require a constant human attention. As a second contribution, VERGILIUS integrates a trace analyzer that extends the IMPORTANT framework [1] to account for urban realistic proagation in the metric computation and further extends the set of network and connectivity metrics.

The reminder of this paper is organized as follows: in section 2 we introduce VERGILIUS, our urban mobility and propagation toolbox, in section 3 we show how the different metrics and parameters affect the network performance in an urban scenario, and finally in section 4 we conclude our

*This work was partially supported by Microsoft Research through its PhD Scholarship Program and by the National Science Foundation under Grant # 0520332-NSF-SWARMS and UC Discovery Grant # SEA private Sponsor STMICROLECTRONICS. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the research sponsors

[1]Publius Vergilius Maro was Dante's guide to Hell and the greater part of Purgatory in The "Divine Comedy." We offer our tool to the research community as aid to explore the design space in vehicular ad hoc networks. The software can be downloaded at `http://vehicular.cs.ucla.edu/`

paper.

2. SYSTEM DESCRIPTION

The general functionality of the proposed toolbox is represented in figure 1. The first step performed is to extract the geographic information available from the TIGER database [4] and translate it to a road topology description. Using this description we then build a road traffic pattern that we use to feed the chosen Mobility Simulator (currently CORSIM [3] or SUMO [2]). The Mobility Simulator provides a mobility trace that together with the road topology information is fed into both the Trace Analyzer and the Network Simulator. At the end of this chain we obtain two different sets of metrics: from the Trace Analyzer we obtain metrics that are solely related to the mobility pattern; from the Network Simulator we obtain metrics that are related to the network protocols and are affected by the characteristics of the mobility pattern. In the following sections we describe in detail how each step is performed.

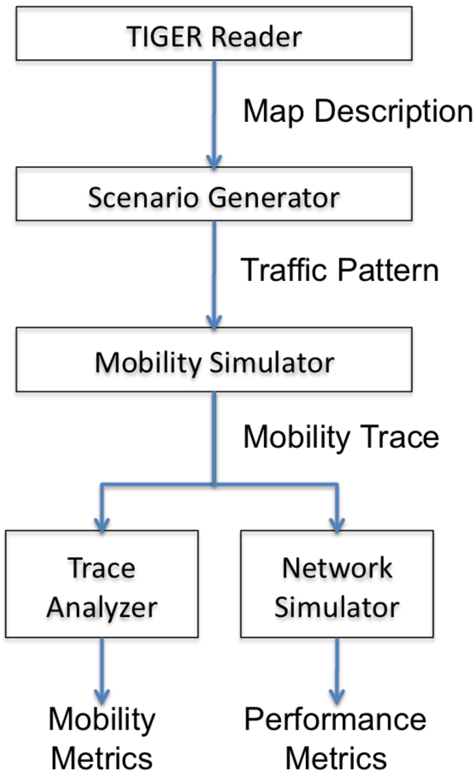


Figure 1: Framework Flow Diagram

2.1 TIGER Reader

The TIGER database [4] contains information about all the geographic lines over the U.S. territory. Geographic lines include roads, railways, county borders, coast lines and several others. Each of this lines is divided into segments that are defined by the two end points and several characteristics. Our concern falls upon the segments that are marked as roads. In the database roads are classified into several categories through the field Census Feature Class Code (CFCC). We use this classification to assign to each road a speed limit and a number of lanes following the schema shown in table 1. This heuristic classification was derived matching the codes to some of the real characteristics of the roads around the Westwood area. Unfortunately TIGER does not include any information about one way roads, therefore we consider all the segments as two way roads. The TIGER Reader module translates the "all segments" data representation into a directed graph made of nodes and edges. Where the nodes are the intersections and the edges are the road segments.

Table 1: Mapping of TIGER feature classes to road characteristics

CFCC	Speed Limit [mph]	Number of Lanes
A1x	65	6
A2x	45	4
A3x	35	2
A4x	25	1
A63	25	1

2.2 Scenario Generator

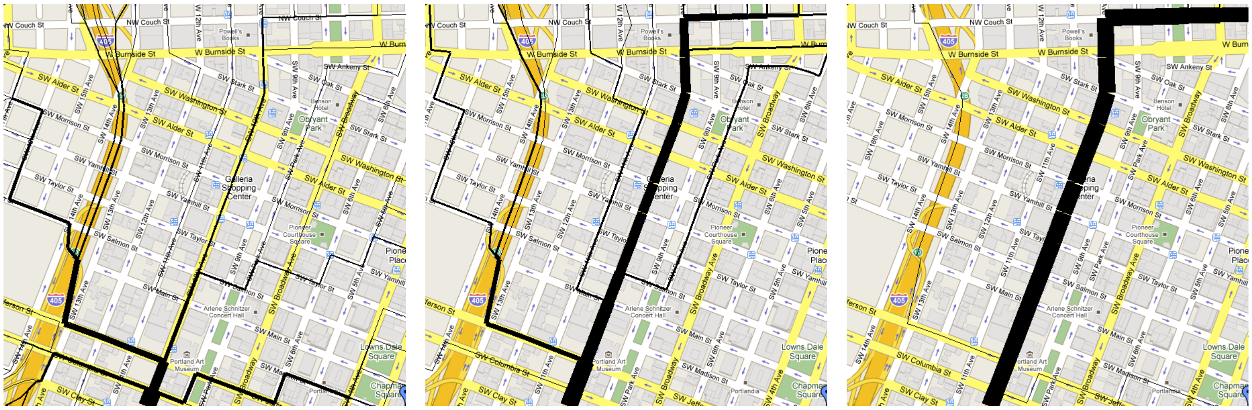
Before describing the functionality of the Scenario Generator we must introduce a new concept: the road traffic pattern. A road traffic pattern is a set of parameters that together define the macroscopic behavior of road traffic over a certain road topology. A road traffic pattern is basically the set of inputs that a mobility simulator needs in order to build the mobility trace. As different mobility simulators require different sets of inputs we define the road traffic pattern as a broader set such that two different simulators fed with the same traffic pattern will produce two mobility traces with the same macroscopic characteristics. Thus, the road traffic pattern can be viewed as the set of primitives that uniquely define the macroscopic traffic behavior no matter which simulator is used to create a realistic traffic from these primitives. The road traffic pattern consists of the following primitives:

1. *Road Topology:* a description of the road network as a directed graph.

2. *Input Flows:* The number of vehicles per unit of time entering from each ingress road segment (i.e. roads placed at the border of the map).

3. *Internal Flows:* The number of vehicles per unit of time that traverses each internal road segment in the map.

4. *Turning Probabilities:* For each intersection, the fraction of vehicles that transition from an incoming road segment to any of the outgoing segments.



(a) $TAF = 0$ (b) $TAF = 5$ (c) $TAF = 100$

Figure 2: Example of path generation from a single source for different values of the Trip Aggregation Factor TAF. The thickness of the segment represents the number of paths traversing it.

5. *Origin Destination Matrix:* Number of vehicles that will depart from any ingress road segment and will travel to ay egress road segment.

Most of the mobility simulators do not require all the listed traffic primitives. In fact CORSIM only requires (1,2,4) and SUMO might be launched either with $(1, 2, 4)$ or (1,5). However we need to include all the items in the definition in order to have a unique description of the road traffic. For example if only the Origin Destination Matrix is given, this may lead to an infinite set of traffic patterns, one for each different set of Turning Probabilities.

Up to now the VANET research community has always run urban simulations either over a single traffic pattern (as realistically mapped to actual measurement) or over a series of random generated traffic patterns. In both cases the simulation looses generality; in fact in the case of a single traffic pattern, although this pattern might be perfectly realistic (in fact it may derive from a real trace), the results are valid only for that specific scenario. In the second case (i.e. a sequence of different patterns) the results although more general do not follow a systematic method, and there is no control on what the scenario could be. Our proposed Scenario Generator is a tool that generates traffic patterns in a controlled fashion, allowing one to fine tuning of the output traffic pattern behavior by manipulating only relatively few parameters as inputs.

In a nutshell the Scenario Generator builds a set of paths on top of the input road topology. The user has control on how to generate this set. The paths always originate from an entry point and end in one of the exit points; entry and exit points are road segments on the border of the map. For each entry it defines the number of paths departing from it and their destinations.

Number of Paths. The user defines the total aggregate input flow rate. This aggregate input flow, F_I, will then be distributed among all the entries based on the importance of the entry road segment. Each class of road will be assigned an incremental weight based on the speed limit as shown in table 2: Where E_{FI} is the exponent given to the weights and

Table 2: Weight Assignment for entry roads

Speed Limit [mph]	Weight
≥ 65	$4^{E_{FI}}$
$\geq 45, < 65$	$3^{E_{FI}}$
$> 25, < 45$	$2^{E_{FI}}$
≤ 25	$1^{E_{FI}}$

can be input from the user. It is also possible to set E_{FI} to 0 in order to have uniformly distributed weights. Each entry i is then assigned the following number of paths N_{Pi}:

$$N_{Pi} = p * W_i \tag{1}$$

Where W_i is the weight of the i^{th} entry road segment and p is a random variable that follows a Poisson distribution with expected value of λ with:

$$\lambda = \frac{F_I}{\sum_i W_i} \tag{2}$$

Choice of Destination. A destination must be chosen for each path assigned to an entry. As mentioned before all paths end in one of the exit points. For each entry we create a list of possible destinations and we sort it in order of descending geometric distance from the entry. The user has control on how to perform the choice either deterministically (choosing always the farthest exit point) or randomly. In order to give to the user control on the way the destinations are chosen, we devised a tunable distribution. Through what we define as Trip Aggregation Factor (TAF) the user can variate the distribution of the random from a uniform distribution towards a Dirac Delta distribution (that coincides with the deterministic choice). The random variable v = number of paths with such distribution is obtained as follows:

$$v = \left| \frac{\sum_0^{2*TAF} u}{2TAF} \right| \tag{3}$$

Where u is a random variable uniformly distributed over the interval $[-0.5, 0.5]$. In figure 2.2 are shown the paths generated from a single source using 3 different values of TAF: 0, 5 and 100 respectively in 2(a), 2(b) and 2(c). In the three figures the thickness of the line represents the number of paths that traverse a certain road segment. As anticipated as TAF grows the paths are more and more directed towards the opposite side of the map.

Path Construction. Once the origin and destination have been set we need to build up the actual path, that is the sequence of road segments a vehicle will traverse to move from origin to destination. We chose to compute the best path using the well known Dijktsra algorithm. The user has control on what weights should be used for the Dijkstra algorithm. The two possible choices are geometric distance and time to traverse that lead respectively to shortest and fastest path.

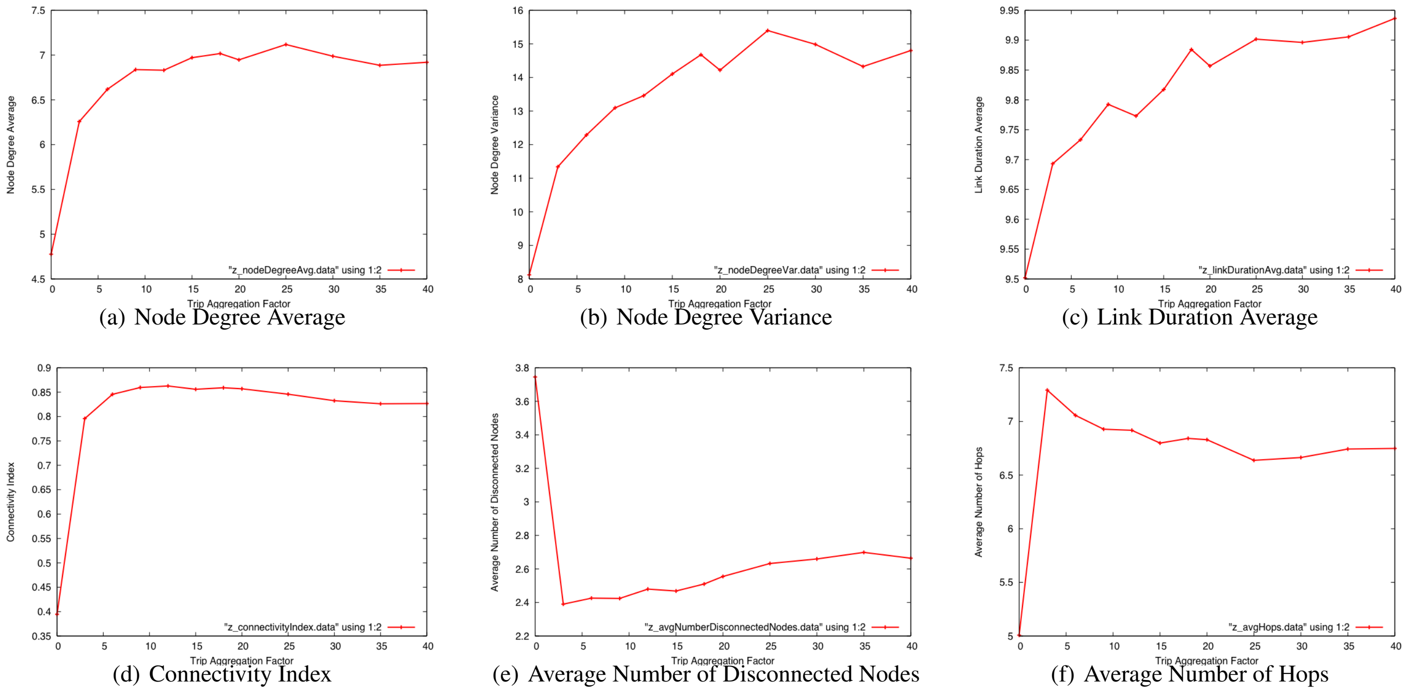
2.3 Trace Analyzer

The Trace Analyzer is a stand alone tool that given a mobility trace and a road topology computes several metrics. This tool intends to be an extension of the IMPORTANT framework [1]. The IMPORTANT framework introduces a set of metrics that fully characterize a mobility trace. The IMPORTANT framework enables the evaluation of the impact on the performance of routing protocols for Mobile Ad-Hoc Networks (MANETs) caused by variations of the mobility pattern that can be expressed by such metrics. The IMPORTANT framework is a very general tool that intends to be valid for all kinds of MANET. Our tool instead focuses only on Vehicular Ad-Hoc Networks (VANETs). Of the several metrics introduced by the IMPORTANT framework we chose to implement only the Node Degree (ND) and the Link Duration (LD). The Node Degree is the number of neighbors each vehicle has in each time period of the simulation and provides a good insight of the density of the network. Together with the average of the ND it is also interesting evaluating its variance that can provide information on how uniform is the density of vehicles along the map. The Link Duration is the number of time periods that two vehicles are in range of each other without interruptions. Leaving behind simple geometric calculations, we take advantage of a novel propagation model presented in [5]. In particular, we compute the path loss between each couple of vehicles using the algorithm presented in [6]. We consider the two vehicles as neighbors if the attenuation is smaller than $97dB$. Using this neighboring information we are able to build the connectivity graph that allows us to evaluate network partitions, path lengths and node disconnections. On top of the connectivity graph we compute for each node the shortest path to each other node in the network using Dijkstra's algorithm. This information allows us to introduce a new set of metrics:

- **Connectivity Index (CI):** defined as the average portion of the network that is reachable from each vehicle in the mobility trace, regardless of the path length. CI provides a better insight onto the partitioning of the network, i.e. a low CI corresponds to a very partitioned network.
- **Average Number of Disconnected Nodes (AD):** defined as the average number of nodes that are disconnected from the network, i.e. that have no neighbors. It provides information on how often there are vehicles that are separated from the rest of the network.
- **Hops Distribution Function (HDF):** defined as the average portion of network that is reachable given a maximum number of hops. Provide information about the stretch of a network, i.e. what is the average distance in hops needed to traverse the network.
- **Average Number of Hops (AH):** defined as the average number of hops needed to reach all nodes in the network.

3. EVALUATION

In this section we evaluate the impact of scenario changes over the connectivity characteristics of the resulting mobility trace. We chose to use a 750x750 meters map of downtown Portland (WA) centered in (Lat: 45.520960 N Lon: 122.679824 W). We fixed the aggregate input flow to 6000 vehicles per hour, and the weight exponent $E_{FI} = 2$ (see section 2.2). We then range TAF in the interval $[0, 40]$. As TAF grows the paths that are generated from a single source will be more aggregated meaning they will end in destinations that are closer to each other. The path aggregation causes the vehicles to form streams that cross the map traveling on prevalently the same roads creating (road traffic) congestion. This congestion reflects onto the connectivity metrics as the vehicles will travel together for longer time. One can observe the effect of such congestion and aggregation in figure 3. More specifically we can observe that the Node Degree Average, the Connectivity index and the Average Number of Hops almost double up as shown respectively in in figure 3(a), 3(d), and 3(f). Moreover we observe a sensible increase of the Link duration (figure 3(c)) and a severe drop of the Number of Disconnected Nodes (figure 3(e)). Another interesting phenomenon is the increase of the Node Degree Variance (figure 3(b)). The Node Degree Variance, as mentioned in section 2.3, is a good indicator of the order of the changes in density of vehicles along the map. In fact, the path aggregation gathers most of the vehicles onto some roads, leaving some areas of the map with just a few vehicles. This causes great variations of density along different areas of the map resulting in a higher Node Degree



(a) Node Degree Average (b) Node Degree Variance (c) Link Duration Average

(d) Connectivity Index (e) Average Number of Disconnected Nodes (f) Average Number of Hops

Figure 3: Variation of the mobility metrics for a growing TAF

Variance. As evident from figure 3, the effect of TAF saturates for values larger than 10. To further increase the vehicle aggregation/congestion it is possible to increase the input weight exponent E_{FI}, concentrating most of the traffic onto the more important roads.

4. CONCLUSIONS

With this paper we presented VERGILIUS: a new toolbox for generating and analyzing urban vehicular scenarios. VERGILIUS enables more general and more reliable simulation studies for VANETs. VERGILIUS provides two innovative tools: a finely tunable mobility Scenario Generator to systematically explore the design space of protocols for VANETs and a Trace Analyzer to analyze and characterize urban mobility traces. Using VERGILIUS we generated and analyzed a set of mobility traces evaluating the impact of the newly introduced Trip Aggregation Factor (TAF).

5. 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 Adhoc NeTworks. In *IEEE INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, 2003.

[2] Deutsches Zentrum für Luft-und Raumfahrt e.V. (DLR). Sumo - simulation of urban mobility. `http://sumo.sourceforge.net/`.

[3] University of Florida. Traffic software integrated system - corridor simulation. `http://mctrans.ce.ufl.edu/featured/tsis/`.

[4] Census. Topologically integrated geographic encoding and referencing system. `http://www.census.gov/geo/www/tiger/`.

[5] Q. Sun, S. Y. Tan, and Kah C. Teh. Analytical formulae for path loss prediction in urban street grid microcellular environments. *IEEE Transactions on Vehicular Technology*, 54(4):1251–1258, July 2005.

[6] Eugenio Giordano, Enzo De Sena, Giovanni Pau, and Mario Gerla. Vergilius: A scenario generator for vanets. Technical Report TR090012, UCLA Computer Science Department, 2009.