An Agent Based Simulation Model for Warning Messages Dissemination in a Vehicular Ad hoc Network

A. Madani

Département de Mathématiques & Informatique
Faculté des sciences
Université Chouaïb Doukkali
El Jadida, Morocco
madaniabdellah@gmail.com

Abstract— Since the safety on roads has become a main concern for both governments and car manufacturers in the last twenty years, number of applications into the domain of vehicular communication is proposed. Vehicular Ad hoc Networks, or VANETs, fall in this category. In this paper we propose an agent based simulation model for warning messages dissemination to study the impact of self-pruning broadcast algorithm on the dissemination of warning messages between vehicles. The results are then compared to those provided by the basic simple flooding broadcast algorithm. Simulation results show that among the key factors affecting warning messages dissemination are: (i) the transmission range, (ii) the density of vehicles in the road, and (iii) the selected broadcasting method.

Keywords—Warning Messages Dissemination, VANET, Blind flooding algorithm, Self-pruning algorithm, Cellular Automaton

I. INTRODUCTION

During the last decades, the evolution of wireless technologies has allowed researchers to design communication systems where vehicles participate in the communication networks. Thus, new types of networks, such as Vehicular Ad hoc Networks (VANETs), have been created to facilitate the communication between vehicles themselves and between vehicles and infrastructures.

Under a number of different terms such as Inter Vehicles Communication (IVC) or Wireless Access in Vehicular Environment (WAVE), the VANET is a special case of the Mobile Ad hoc Network (MANET) in which communication nodes is mainly vehicles. The main objective of VANETs is to provide a robust network between mobile vehicles so that cars can communicate to each other (Vehicle-to-Vehicle communication) or with other fixed road-side equipments (Vehicle-to-Infrastructure communication) to get services or to disseminate messages.

An important factor in warning messages dissemination in VANETs is the selected broadcast scheme. In VANETs, intermediate vehicles act as relays to support end-to-end vehicular communications. Most of the major broadcasting methods rely on flooding for disseminating messages between vehicles. The blind flooding [1] method is a simple and straightforward approach for broadcasting without global information. This approach starts with a source node broadcasting a packet to all neighbors. Each of those neighbors in turn rebroadcast the packet exactly one time and this continues until all reachable network nodes have received the packet. The blind flooding will generate many transmissions. The redundant transmissions may cause the well known broadcast storm problem [2], in which redundant packets cause contention and collision.

Many broadcast algorithms besides blind flooding have been proposed to solve the broadcast storm problem in ad hoc network [3, 4, 5, 6, 7, 8]. The self-pruning approach [9] is an effective method in reducing broadcast redundancy. This approach requires that each node have knowledge of its one-hop neighbors, which is obtained via periodic “Hello” packets. A node includes its list of known neighbors in the header of each broadcast packet. A node receiving a broadcast packet compares its neighbor list to the sender's neighbor list. If the receiving node would not reach any additional nodes, it refrains from rebroadcasting; otherwise the node rebroadcast the packet.

This paper aims to create a flooding protocol-based system by which vehicles transmit information about traffic conditions to help other drivers to take adequate decisions. Such systems can be used to change our way to drive, improve road safety, and help emergency services. Simulation results show that among the key factors affecting warning messages dissemination are: (i) the transmission range, (ii) the density of vehicles, and (iii) the selected broadcasting method. The paper is organized as follows. Section II presents some features that enhance the modeling expressiveness of the platform of NetLogo. The description of our problem is introduced in section III. Section IV summarizes the results of our simulation. Finally, the conclusion and some future works are presented in section V.

II. NETLOGO: A MODELING TOOL

NetLogo is an excellent tool for rapid prototyping and initial testing of multi-agent systems, particularly suited to systems with agents situated and operating in a restricted space, as well as an excellent animation tool of the modeled system [10]. It is designed for both research and education and is used across a wide range of disciplines and education levels. NetLogo is a freely tool created by Uri Wilensky at the center of Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, Illinois. NetLogo has a simple syntax and well-thought out system to create complex models. It consists of a world and four types of agents which are turtles, patches, links and the observer.

The world is the environment of the turtles, patches and links, and it can be either 2D or 3D. Turtles are the autonomous agents moving around the world. Patches are the squares composing the ground of the world. Turtles and patches have coordinates and color attributes. Communication between turtles is provided over links. Links do not have coordinates. The observer simply controls the experiment, in which turtles and patches actively participate. Both patches and turtles can inspect the environment around them, for example detect the existence of other agents, view the state of their surrounding patches/turtles, and modify the environment. Agents can be organized in groups under a user specified name, and thus agents of different breeds can exist in the simulation world. Probably the feature that most greatly enhances the modeling expressiveness of the platform of NetLogo is the fact that each patch and turtle can have its own user-defined variables: in the case of patches this allows modeling complex environments by including an adequate number of variables that describe it sufficiently and in the case of turtles it simply means that each agent can carry its own state, stored again in a number of user defined variables.

The NetLogo programming language is simple and expressive and has a rather functional flavor. There is an extensive set of programming primitives, good support for floating point mathematics, random numbers and plotting capabilities. Programming primitives include for example, commands for “moving” the turtles on the grid, commands for environment inspection (i.e. the state of other turtles and patches), classic programming constructs (branching, conditionals, repetition), etc. The main data structure is lists, and the language supports both functions, called reporters, as well as procedures. Monitoring and execution of the agents is controlled by the observer entity that “asks” each agent to perform a specific computational task. The programming environment also offers GUI (Graphical User Interface) creation facilities, through which custom visualizations of the studied multi-agent systems can be created with particular ease.

III. DESCRIPTION OF THE PROBLEM

In this section we describe how the driver warning system that we proposed operates, as well as the technologies and protocols involved. In our traffic safety system, vehicles may send warning messages to other vehicles in order to prevent collisions or to ask for emergency services. We consider that a sensor is located on the road and periodically sends the warning messages (figure 1). When a vehicle receives a broadcast message, it stores and immediately forwards it by re-broadcasting the message. Warning messages should be propagated to all neighbors up to a certain number of hops, and so a flooding-based routing protocol fits our requirements adequately. We pretend that the warning packets sent by a vehicle can be received by all the vehicles in the nearby area, and so this protocol offers the best reliability in terms of coverage.

Our model considers a single lane car traffic that can be modeled as follows. The road is represented as a line of cells of constant length (7.5 meters long) to form one-dimensional cellular automaton. Each cell can be empty or occupied by exactly one vehicle. The cars move from the left to the right on a lane with periodic boundary conditions. At each time step, the system is characterized by the position and the velocity configurations of cars. The velocity represents the number of cells that a vehicle advance in one update and it is represented as an integer between 0 and v_{max}, where v_{max} is the speed limit. Let x(i,t) and v(i,t) denote the position and the velocity of the ith car at time t, respectively. The number of empty cells in front of the ith vehicle at time t is denoted by $g(i,t) = x(i+1,t) – x(i,t) - 1$. g(i,t) is the gap between the ith car and the preceding one.

Starting from a given configuration at a time t, the configuration at the next time (t+1) can be obtained by applying four simple rules of the Nagel-Schreckenberg model (NaSch) [11], as follows:

1. Acceleration : $v(i,t+1) \leftarrow \min(v(i,t) + 1, v_{max})$
2. Deceleration : $v(i,t+1) \leftarrow \min(v(i,t+1), gap(i,t))$

3. Noise : v(i,t+1) ← max(v(i,t+1) – 1, 0), with probability p
4. Motion: x(i,t+1) ← x(i,t) + v(i,t+1)

These four rules can be interpreted as follows: the driver tends to drive as fast as possible without exceeding the maximum velocity (rule 1), but he must decelerate to avoid collision (rule 2). The rule 3 called also the randomization rule, takes into account the different behavioral patterns of the individual drivers, especially nondeterministic acceleration and overreaction while slowing down. The rule 4 moves the vehicle according to its current velocity.

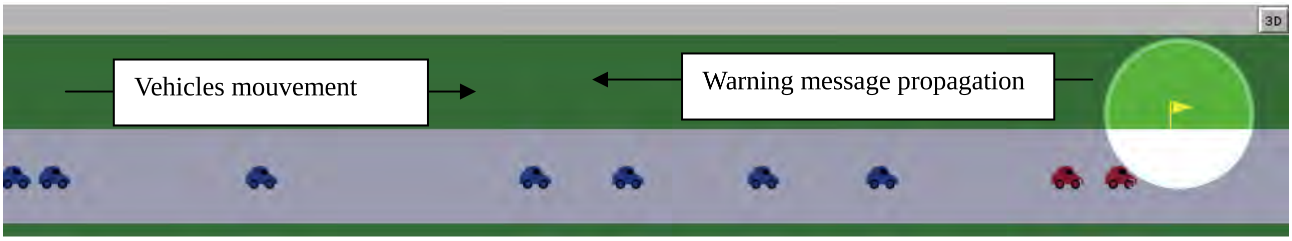

Figure 1. Representation of the warning message propagation

Figure 1 shows the simulated road. The yellow flag (at the extreme right of the figure) represents the road-side sensor while red cars represent vehicles that received the warning message and must send it to other vehicles in the transmission range. The propagation of the warning message is performed in the opposite direction of vehicles movement.

IV. Simulation Results

Simulation results presented in this paper were obtained using the NetLogo tool. Before going on, we would like to describe our standard simulation set-up for the following observations. We simulated a system of length L=1000 cells with closed boundary conditions, i.e. the traffic is running in a loop. We start with random initial conditions. N cars are randomly distributed on the lane around the complete loop with an initial speed taking a discrete random value between 0 and v_{max}. Each vehicle on the road is equipped with the adequate wireless capability. Since the system is closed, the average density =N/L remains constant with time. Next, we update the individual car velocities and positions in accordance with rules of the NaSch model. Each density is simulated for T=2000 time steps, of which the first half (1000) were discarded to let transients die out and for the system to reach its asymptotic steady state. Table 1 shows the parameters used in this paper.

Table 1. Parameters selected in the simulation

Length of each cell (m)	7.5
Length of each car (cell)	1
Total number of cells	1000
Maximum velocity (cell)	5
Randomization probability	0.3

Figure 2 reports the average probability of informing all vehicles (P_{info}) in the road as a function of the density of the road ρ for the two broadcasting methods: the simple flooding method and the self-pruning method. The transmission range is fixed to 8 cells. As expected, the probability P_{info} increases with the density for the two methods. This result is expected, since increasing the number of vehicles in the road increases the number of neighbors and therefore the number of active connections between vehicles.

On the other hand, for the two broadcasting methods, the simple flooding method is the one who shows more informing vehicles. This is due to the fact that the self-pruning method requires the least number of rebroadcast than the simple flooding method. But the two curves tend to the same value when the road becomes dense.

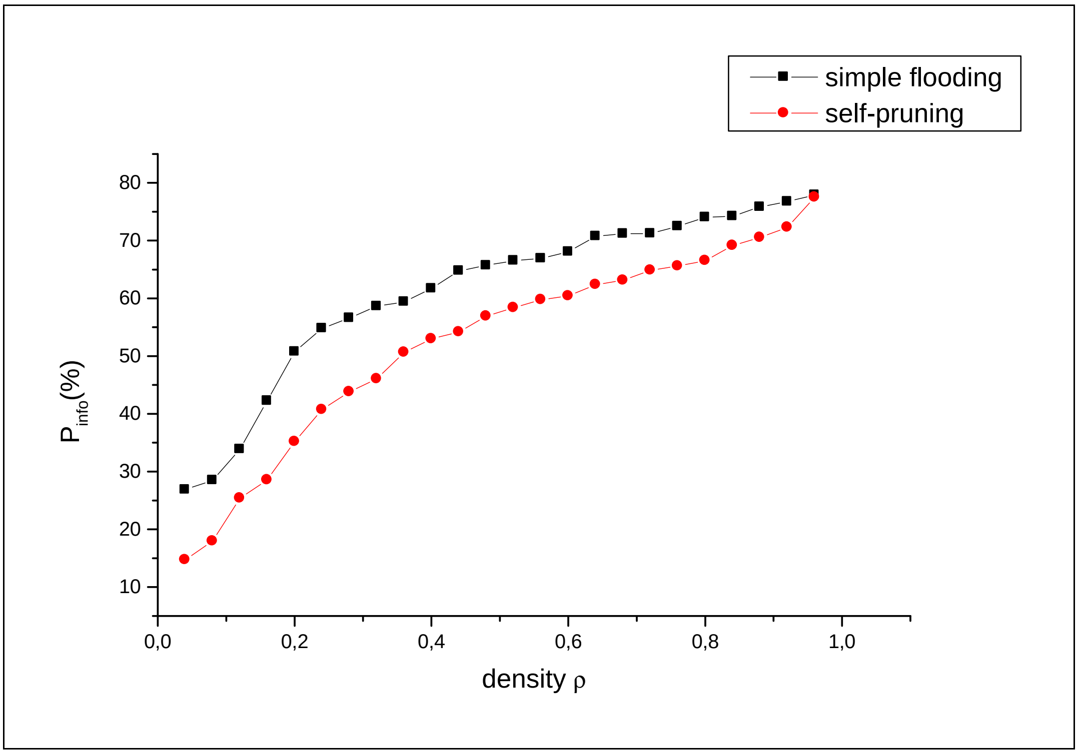

Figure 2. Average probability of informing all vehicles (P_{info}) vs. the density ρ for a fixed transmission range

Figure 3 shows the simulation results of the probability P_{info} obtained with the self-pruning method for different values of transmission range. It can be seen clearly that the transmission range has an impact on the probability P_{info}. As soon as the transmission range grows, the probability P_{info} increases. This is obvious since for a large value of the transmission range, a vehicle is more likely to have more neighbors and therefore more active connections with other vehicles or with other fixed road-side equipments.

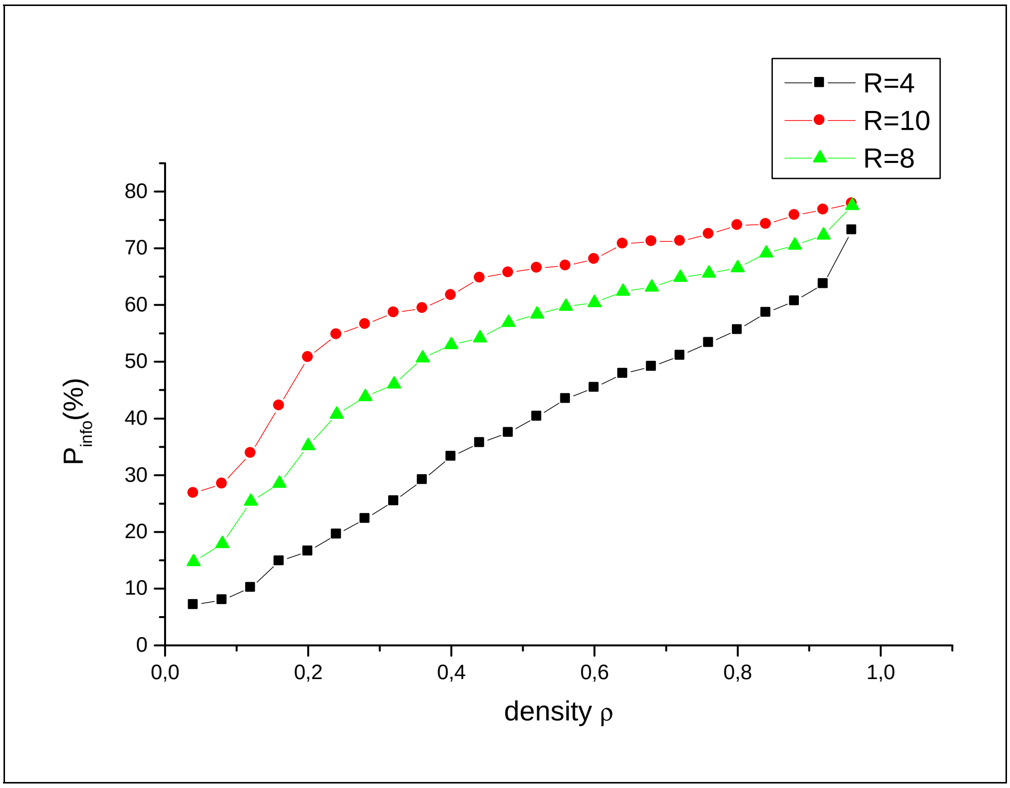

Figure 3. Average probability of informing all vehicles (P_{info}) obtained with self-pruning method vs. the density ρ for different transmission ranges

V. CONCLUSION AND FUTURE WORKS

In this paper we have studied the average probability of informing all vehicles (P_{info}) in a road using two different methods of broadcasting: the simple (blind) flooding method and the self-pruning method. Our results show that the communication between mobile vehicles of a vehicular ad hoc network is greatly affected by several factors, such as, the way by which vehicles communicate with each other or with road-side equipment and the transmission range.

Despite the existence of several other agent modeling tools, the main advantage of NetLogo is the ease of programming. The language is very intuitive and since it is specifically designed for Agent Based Modeling, the user needs only to program agent behavior, not the agent itself. This makes it possible to iteratively change, test and improve the model very quickly.

In our approach we have used only one roadside sensor to initiate the communication between vehicles in the road. The V2V (vehicle-to-vehicle) communication using one-hop forwarding is not effective in the case of low density or sparsely connected vehicular networks. Thus, in future work, we expect using several road-side sensors to allow multi-hop forwarding in order to propagate the messages or signals effectively.

REFERENCES

[1] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, “Flooding for reliable multicast in multi-hop ad hoc networks”, In proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIALM), pp. 64 – 71, 1999

[2] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, “The Broadcast Storm Problem in a Mobile Ad Hoc Network,” Proc. MOBICOM ’99, pp. 151-162, Aug. 1999

[3] K.M. Alzoubi, P.J. Wan, and O. Frieder, “New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks,” Proc. Hawaii Int’l Conf. System Sciences ’35, 2002.

[4] G. Calinescu, I. Mandoiu, P.J. Wan, and A. Zelikovsky, “Selecting Forwarding Neighbors in Wireless Ad Hoc Networks,” Proc. ACM Int’l Workshop Discrete Algorithms and Methods for Mobile Computing (DIALM ’01), pp. 34-43, Dec. 2001.

[5] H. Lim and C. Kim, “Flooding in Wireless Ad Hoc Networks,” Computer Comm. J., vol. 24, no. 3-4, pp. 353-363, 2001.

[6] W. Peng and X.C. Lu, “On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks,” Proc. First Ann. Workshop Mobile and Ad Hoc Networking and Computing, MOBIHOC, pp. 129-130, Aug. 2000.

[7] Qayyum, L. Viennot, and A. Laouiti, “Multipoint Relaying for Flooding Broadcast Message in Mobile Wireless Networks,” Proc. Hawaii Int’l Conf. System Sciences ’35, Jan. 2002.

[8] Stojmenovic, S. Seddigh, and J. Zunic, “Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks,” IEEE Trans. Parallel and Distributed Systems, vol. 13, no. 1, pp. 14-25, Jan. 2002.

[9] H. Lim and C. Kim., “Multicast tree construction and flooding in wireless ad hoc networks”. In Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), 2000

[10] Vidal, J.M., Buhler, P., Goradia, H.: The past and future of multiagent systems. In: Proceedings of 1st AAMAS Workshop on Teaching Multi-AgentSystems (2004).

[11] K. Nagel and M. Schreckenberg, “A cellular automaton model for freeway traffic”, Journal of physical I France, Vol. 2., pp. 2221-2229, 1992