Absolute Priority for a Vehicle in VANET

Rostam Shirani, Faramarz Hendessi, Mohammad Ali Montazeri,
and Mohammad Sheikh Zefreh

Department of Electrical and Computer Engineering, Isfahan University of Technology,
Isfahan, 84156-83111, Iran
{rostam_sh,sheikhzefreh}@ec.iut.ac.ir,
{hendessi,montazer}@cc.iut.ac.ir

Abstract. In today's world, traffic jams waste hundreds of hours of our life. This causes many researchers try to resolve the problem with the idea of Intelligent Transportation System. For some applications like a travelling ambulance, it is important to reduce delay even for a second. In this paper, we propose a completely infrastructure-less approach for finding shortest path and controlling traffic light to provide absolute priority for an emergency vehicle. We use the idea of vehicular ad-hoc networking to reduce the imposed travelling time. Then, we simulate our proposed protocol and compare it with a centrally controlled traffic light system.

Keywords: Intelligent Transportation System, Traffic Light Control, Shortest Path Algorithm, Vehicular Ad-Hoc Network.

1 Introduction

Two important usages of Intelligent Transportation System (ITS) are shortest path (SP) finding and traffic light control (TLC). In TLC, the idea is to control the traffic lights in order to optimize a desired parameter. In this paper, our goal is to minimize the travel time of a distinguished vehicle from a source to a destination. The advantage of this protocol is when we want to provide an Absolute Priority for a vehicle. In absolute priority, all of the available network capabilities are used to impose the lowest possible travel time to an emergency vehicle such as an ambulance.

For the rest of the paper, we use ambulance as a symbolic name for each vehicle that needs absolute priority in an urban environment. Dijkstra [1], the most common SP algorithm in VANET scenario, is used to find SP. In order to have TLC, in [2], the idea is to use Geographical Information System (GIS). Huang proposed a design strategy based on statechart for TLC [3]. Furthermore other researchers have used the idea of Intelligent TLC [4]. In this paper, we propose a decentralized algorithm with simultaneous use of SP and TLC based on packet structure of Dynamic Source Routing (DSR) [5]. Our proposed structure, which acts in an emergency situation, could be used as a complement for a central traffic control system.

Our proposed protocol consists of two algorithms. One of these algorithms tries to estimate the cost of different paths towards a destination without relying on GIS or any other central data base. After that, it is necessary to have an efficient forwarding mechanism for addressing MAC issues. The aim of this mechanism is to send packets

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 955–959, 2008.

with the lowest possible delay. And finally, the most important algorithm is the one which keeps the traffic light green to prepare absolute priority for the ambulance.

2 Protocol Description

We consider that each vehicle has an ID, and ambulance ID is recognizable for traffic lights. We assume that vehicles are equipped with GPS and Digital Road Map. Therefore, it is not very hard to calculate the physical distance. But decentralized calculation of vehicles' density is more challenging. However, our method is able to intelligently recognize SP (the path with the least imposed travelling time). The packet structure is similar to a usual DSR packet which includes three additional parts: A critical tag used by an ambulance to show emergency situation, a tag which is used as a sign for other vehicles to ask them whether or not insert their characteristics, and a table including vehicle's velocity, vehicle's and street's ID, and vehicle's position based on GPS information.

In the first step, the algorithm should find the SP. The ambulance starts sending requests towards destination. Because transmission range is limited, a multi-hop mechanism should be applied. All the intermediate nodes insert their velocity, their ID, and their current street's ID in the receiving packet and then forwards the packet. When the packet reaches to the destination, it should travel back towards source. In this returning trip, intermediate vehicles act as relays. When the packets of different paths returned back to source, the cost values will be estimated.

In our design, cost is defined as travelling delay (the time that is needed for a vehicle to travel from a source to a destination); this time is different for different paths. The more the delay of a path is, the more the cost of that path will be. For estimating travel time, two parameters are needed: physical distance from source to destination, and average speed in the path. The physical distance is calculated by GPS.

In order to estimate the speed, we use an averaging mechanism. We assume that each vehicle inserts its current speed in the packet. Hence, by averaging over speeds of vehicles which are in a specific street, the cost of that street is estimated. Assume that C_i is the cost value of the street S_i. Also, L_i is the length of the street S_i. In each street several vehicles cooperate to form a multi-hop scenario, $V_i = \frac{1}{K}\sum_{k=1}^{K} V_i^k$ is the average velocity of vehicles which cooperates in message forwarding in street S_i. K is number of vehicles in S_i which participates in data forwarding. When the packet comes back to source node (ambulance), it is easy to calculate delay of different paths. Delay of the m^th^ path is shown in (1).

$$S_m \Rightarrow \begin{cases} L_m & \text{Distance from digital road map} \\ V_m & \text{Velocity from averaging} \end{cases} \Rightarrow T_m = \frac{L_m}{V_m} \tag{1}$$

As it was mentioned before, cost value of each path is defined as related delay of that path. Therefore T_1, T_2... T_N could be used as cost values. In other words, $C_i = T_i$ is the cost value of street S_i. A summation over the streets' delay of a path gives the cost of that path. Therefore, Dijkstra algorithm uses these cost values for finding SP. In the following algorithm, the task of source is shown.

Step 1- setting the critical bit
Step 2- setting velocity tag
Step 3- sending a request towards destination coordination (intermediate vehicles insert their velocity in the corresponding part of the packet towards destination)
Step 4 -waiting for arrival of replay packet
Step 5-averaging vehicles' velocities with the same street ID
Step 6-calculating costs of different streets
Step 7- running Dijkstra algorithm

Intermediate nodes are used both for finding SP and for controlling traffic light. In SP finding, intermediate nodes should recognize to add their current velocity, their ID, and their street ID. But in the returning path, they act as relay. An important point about all of the above applications is that a handshaking mechanism is needed for choosing next hop. The purpose of handshaking is to choose the furthest vehicle in the transmission range of the vehicle which carries information now. In figure 1, imagine a packet in vehicle A is ready for transmission. Handshaking mechanism should be able to choose C (the furthest node) as the next relay automatically.

Step1- A inserts its ID, its current velocity, and street ID in packet.
Step 2- A sends the request packet.
Step 3- B and C receive the request packet.
Step 4- B sends an Ack which contains its location, C sends an Ack which contains its location.
Step 5- A receives B's and C's Ack.
Step 6- B and C receive each others' Acks.
Step 7- A sends an Ack which contains C's ID; C has been chosen as the next relay.
Step 8- C transmits the packet, B drop the packet.

3 Simulation

We used SUMO [6], and NS2 [7] as our simulation tools. We used SUMO to simulate semi-real traffic patterns. In this work at first, we ran SUMO to gain the traffic patterns. Based on each traffic pattern, delay of centrally controlled traffic light was calculated by SUMO. We evaluated the delay of multiple hops for each link by ns2. Then Dijkstra algorithm was used. We ran our TLC algorithm in SUMO for the same traffic pattern used in centrally controlled traffic light (centrally controlled traffic light is a traffic light without wireless capabilities).

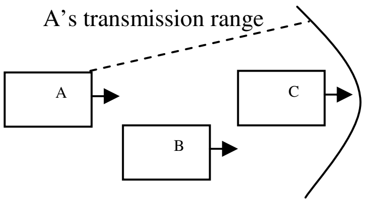

Fig. 1. MAC mechanism chooses C as the next relay

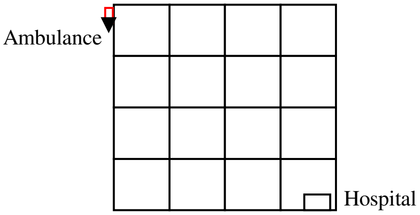

Fig. 2. Simulated urban scenario

The ambulance's source and destination is shown in figure 2. Other vehicles in the scenario have completely random movements with random source and destination. Number of vehicles is varied from 100 to10000 (0.5 veh/km to 50 veh/km). Imposed latency on travel time of the ambulance was calculated both for a centrally controlled traffic light system and for absolute priority. Simulation results have been shown in figure 3. Travelling Time Ratio is defined as: $TTR = \frac{Travellig\ \ Time\ \ of\ \ Ambulance}{Ideal\ Travelling\ \ Time}$

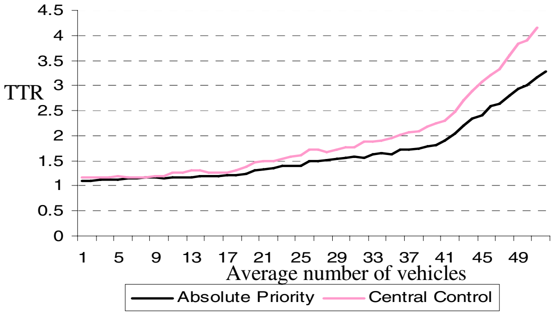

Fig. 3. Travelling Time Ratio versus average number of vehicles

Ideal Travelling Time (ITT) is the time that ambulance can travel from source to destination with maximum velocity which is 60 km/h in our simulation. ITT depends on the length of the SP. This is because increasing number of vehicles imposes upper delay. Also, by increasing average number of vehicles, the distance between these curves will increase. It shows that in rush hours, our method is more beneficial.

4 Conclusion

A decentralized protocol in VANET environment for finding SP and TLC was proposed. Simulation results show the performance improvement of our proposed protocol especially in rush hours. For future works, we try to perform an adaptive SP algorithm to address the dynamic nature of traffic patterns. Also, we want to simulate an environment which all cars announce their delay requirements. Based on these announcements, an intelligent decision is made to optimize the overall delay in a city.

Acknowledgments. This work was supported by Iran Telecommunication Research Center (ITRC).

References

1. Johnson, D., Maltz, D.: Dynamic Source Routing in Ad-Hoc Wireless Networks. In: Proc. ACM SIGCOMM (1996)
2. Bertsekas, D., Gallager, R.: Data Networks, 2nd edn., p. 401.

3. Kumar, R., Reddy, D., Singh, V.: Intelligent transport system using GIS, Map India (2003)
4. Huang, Y.-S.: Design of traffic light control systems using statecharts. The computer journal 49(6) (2006)
5. Wiering, M., Veenen, J.V., Vreeken, J., Koopman, A.: Intelligent Traffic Light Control. Institute of information and Computing and Science, Utrecht University (2004)
6. `http://sumo.sourceforge.net/docs/documentation.shtml`
7. `http://www.isi.edu/nsnam/ns/`