Spatial and Traffic-Aware Routing (STAR) for Vehicular Systems

Francesco Giudici and Elena Pagani

Information and Communication Department,
Università degli Studi di Milano, Italy
{fgiudici, pagani}@dico.unimi.it

Abstract. In this paper, we propose a novel position-based routing algorithm for vehicular ad hoc networks able to exploit both street topology information achieved from geographic information systems and information about vehicular traffic, in order to perform accurate routing decisions. The algorithm was implemented in the NS-2 simulator and was compared with three other algorithms in the literature.

1 Introduction

Progresses in wireless technologies and decreasing costs of wireless devices are leading toward an increasing, pervasive, availability of these devices. Recently, research took interest in possibilities opened by equipping vehicles with wireless devices. Vehicles carrying wireless devices are able to connect with one another in an ad hoc mobile network. These systems can be useful for several distributed applications, ranging from vehicular safety, to cooperative workgroup applications and fleet management, to service retrieval (e.g., availability of parking lots), to entertainment for passengers. Several problems are still to be solved: a main issue is to design effective and efficient routing algorithms appropriate for the characteristics of these systems. A promising approach seems to be using *position-based* routing: routing is performed basing on the current geographic position of both the data source and destination.

In this paper, the novel *Spatial and Traffic-Aware Routing* (STAR) algorithm is proposed, that overcomes drawbacks of other solutions proposed in literature.

2 System Model

In this work, we consider *Vehicular Ad Hoc Networks* (VANETs). As in Mobile Ad Hoc Networks (MANETs) [1], devices – in this case, vehicles – are equipped with wireless network interface cards. Nodes are required to have unique identifiers. The network topology dynamically changes as a consequence of vehicle movements, possibly at high speed. Each vehicle is responsible for forwarding data traffic generated from or addressed to other vehicles. The network is completely decentralized: vehicles have no information on either the network size –

L.T. Yang et al. (Eds.): HPCC 2005, LNCS 3726, pp. 77–86, 2005.

in terms of number of nodes involved – or topology. Two vehicles are said to be *neighbors* if they are in communication range. Differently from MANETs, in VANETs power saving is not of concern. Each vehicle can exploit a *Global Positioning System* (GPS) [2] to determine its own position. A GPS navigation system allows to obtain information about the local road map and the vehicle's direction of movement. For the use of STAR, digital road maps are translated into graphs, with crossroads as vertexes and streets as edges.

According to all currently proposed wireless routing algorithms, vehicles periodically exchange network-layer *beacon* messages, allowing each node to discover the identities and the positions of its own neighbors.

In VANETs, nodes are addressed through their position rather than through their network address. When a vehicle has data to send, it can discover the current position of the receiver by exploiting a *location service*. Several location services have been proposed in the literature [3,4,5,6,7]. We do not make any assumption about the location service used.

In this paper a *pure* wireless ad hoc network is considered, without any access point connected to a fixed network infrastructure. Different mobility scenarios are possible. Often, in the literature, a random-waypoint model is assumed. This model is not able to capture the peculiarities of vehicular movements. In real scenarios, vehicle movements are constrained by roads. In this article, a city mobility model is assumed, along with a Manhattan street topology.

3 Spatial and Traffic-Aware Routing (STAR)

STAR approach to vehicular routing problem is quite different from other position-based routing algorithms. Other existing algorithms [10] may fail in case they try to forward a packet along streets where no vehicles are moving. Such streets should be considered as "broken links" in the topology. Moreover, a packet can be received by a node that has no neighbors nearer to the receiver than the node itself; in this case, the problem of a packet having reached a *local maximum* arises. These problems can be overcome to some extent knowing the *real* topology, that is, by trying to use for packet forwarding only streets where vehicular traffic exists. To reach this objective the STAR algorithm is organized in two layers (fig.1): a lower layer that manages the gathering and exchange of information about network status and a higher layer for the computation of paths. As network status we mean the actual distribution of vehicles along streets. Status knowledge should not concern the whole network: collecting and exchanging information about topology can be expensive, and on the other hand this information is highly volatile due to node mobility. The higher layer takes in charge the route computation on the network topology discovered by the lower layer. Some reference points (*Anchor Points*, or APs for short) on the streets traversed by the computed routes are taken; packets are forwarded from one AP to the successive. It is convenient to compute only a *partial* path to approach the destination position by determining only a subset of Anchor Points. When a packet arrives to the last AP computed for it, the node responsible for

forwarding takes in charge the characterization of the next APs. Partial successive computation of the path has a threefold advantage: (*i*) the size of packet header is fixed; (*ii*) the computation of subsequent APs is done exploiting more updated information about vehicular traffic distribution; (*iii*) subsequent APs can be computed exploiting updated information about the current position of the destination. In the following subsections we explain the functionalities deployed at each layer. For more details, interestered readers may refer to [12].

3.1 Vehicular Traffic Monitoring

We are interested in detecting two extreme situations: the presence of an high number of vehicles - queues - or the total absence of them. Streets with queues of vehicles should be preferred, as they provide several alternatives for packet forwarding, thus minimizing the risk of a packet reaching a local maximum. By contrast, the routing algorithm *must* avoid streets where vehicles are not present, because packets cannot for sure be routed over them as long as their status does not change.

Monitoring and propagation of vehicular traffic conditions are performed through the exchange of network-level *beacons* (fig.1), carrying observations of node neighborhoods. The observations are maintained in data structures managed by the *traffic monitoring* module.

A node maintains the position of its neighbors in the `neighbors-table`. Node neighborhood is discovered via the beacons. In the `presence vector (PRV)` a node maintains four node counters, which represent the number of neighbors it has toward cardinal points computed dividing the node cell into sectors as

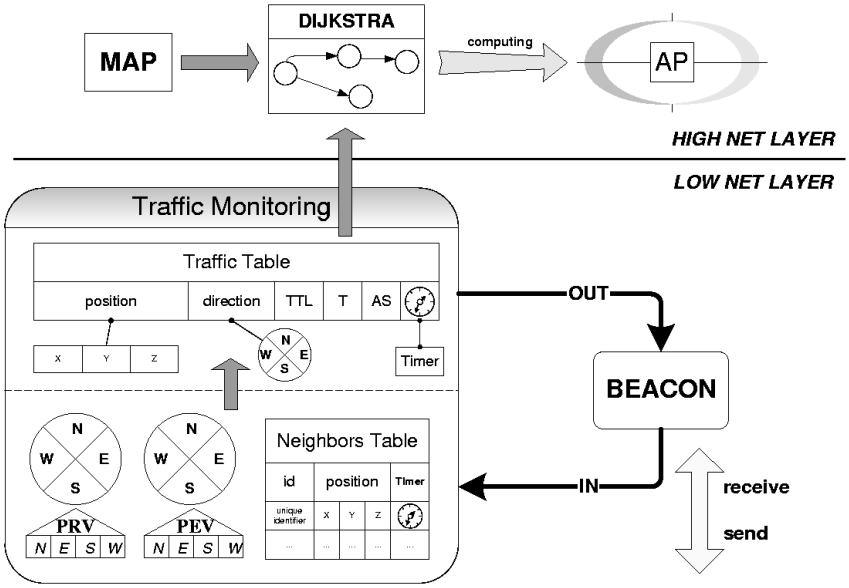


Fig. 1. Functional architecture for the STAR algorithm

shown in fig.1. Each counter is incremented when a neighbor is discovered in the corresponding direction and decremented when a neighborhood information is not refreshed for a certain time.

When a PRV counter exceeds a parameter highPR, a high concentration of vehicles exists in the corresponding direction. By contrast, an elements of PRV below parameter lowPR indicates scarce vehicular traffic along a street in the corresponding direction of the node. When one of these situations occurs, the modification of a related element in the persistence vector (PEV) is triggered. PEV has four elements as PRV. Each element can be in one of three different conditions: it could be in a *reset state* (a value equal to 0), or in a *growing state* (value > 0), or in a *shrinking state* (value < 0). In the event of a PRV counter exceeding highPR, if the corresponding PEV element is in either *reset* or *growing* state, then it is incremented; otherwise is set to *reset state*. On the other hand, in the event of a PRV element below lowPR, if the corresponding PEV element is in either *reset* or *shrinking* state then it is decremented; otherwise is set to *reset state*. PEV is used to register critical situations only when they last for a long time. When the value of an element of PEV goes out of a range [lowPE, highPE], then the information about vehicular traffic stored in the element is recorded in the traffic-table and the value of the PEV element is reset. The use of PEV is necessary to guarantee that a temporary abnormal condition is not registered, but if it lasts then it is registered in the traffic-table.

Each node has a traffic-table. Each entry in this table has five fields: position, direction, traffic bit (Tbit), already-sent bit (ASbit) and Time-To-Live (TTL). The position indicates the coordinates of the node when it first notices an anomalous vehicular traffic situation. The direction is the direction in which the traffic anomaly is taking place with respect to position. Tbit specifies the type of traffic (high or low). ASbit records whether a traffic entry has been already propagated to neighbors and TTL is the number of hops the information has to travel. Each entry has an associated traffic-timer. When the timer expires, the entry is removed from traffic-table in order to forget obsolete information about traffic anomalies.

Each node periodically broadcasts to its neighbors a beacon that contains sender identifier, sender coordinates and the vehicular traffic conditions it has in its traffic-table. Broadcasting period is determined by a beacon-timer.

When a node receives a beacon, it registers the position of the sender in its neighbors-table; PRV and PEV are possibly updated as explained before. If one of the elements of PEV becomes either lower than lowPE or higher than highPE then a new entry is added in the traffic-table. The new entry has as position the coordinates of the node, direction equal to the corresponding element of PEV ('N', 'E', 'S' or 'W'), ASbit set to zero and TTL set to a value maxTTL, which determines how far traffic information will be spread. Tbit is set to the appropriate value 'H' or 'L' according to the vehicular traffic condition detected. After updating the information about neighbors, the local traffic-table is augmented with the vehicular traffic information stored in the received beacon. Each traffic entry in the beacon is copied into traffic-table, with TTL value

decremented by one. While updating the `traffic-table`, existing entries must be compared with the information carried by the beacon. In case two matching entries exist in both the beacon and the table, and they have the same `direction` and `position`s differ less than a parameter `traffic information distance (TID)`, then the two entries refer to the same traffic condition. Only the one having the highest `TTL` is kept. This allows to suppress duplicate advertisements, still guaranteeing that abnormal conditions are notified. If `TTL`s are equal, the `traffic-timer` (determining entry expiration time) of the `traffic-table` entry is set to the initial value, and the `ASbit` is set to 0. This occurs when a traffic anomaly persists for a certain time, and is thus advertised more than once. The receiving nodes must refresh the corresponding `traffic-table` entry and re-propagate this information.

When a node's `beacon-timer` expires, a new beacon carrying node's identifier and position is created. Each entry from the `traffic-table` is copied into the beacon only if `TTL` > 0 and `ASbit` $= 0$. Then the `ASbit` is set to 1 to prevent diffusing multiple times a certain information. Finally the beacon is sent.

3.2 Routing and Packet Forwarding

At the higher layer (fig.1), routes are computed *on-demand* exploiting neighbors and vehicular traffic information locally owned. When a source S has a packet to send to a destination D, S builds a *weighted graph* using street map and traffic information. Edges corresponding to streets without traffic have associated a high weight. By contrast, when in a street there is high vehicular traffic, the weight of the associated edge must be decreased to privilege the choice of this street although it could characterize longer paths. As a consequence of vehicles' mobility, weights of the edges are dynamically adjusted; initial edge weights and the mechanism of weight adaptation must guarantee that weights never become negative or null.

Dijkstra's algorithm is applied to the obtained graph in order to find the shortest route. APs are computed along the streets belonging to the route. The packet header includes the destination identifier, the destination position and a limited number of APs: the packet is forwarded with GEOGRAPHIC GREEDY ROUTING [10] from one AP to the other. When the last AP is reached, the node in charge of forwarding the packet will compute other APs toward the destination, until it is reached. In case of routing failure, the *recovery procedure* adopted by STAR consists in computing a new route from the current node, exploiting updated traffic information.

3.3 Dimensioning of Parameters

STAR behavior is controlled through some parameters. The `CROSS RANGE ACCEPT (CRA)` parameter is introduced to enhance STAR traffic detection: if a vehicle is moving along a straight road it does not need to collect information about traffic in (non-existent) lateral streets. If a vehicle is far from the nearest crossroad more than `CRA` meters, then it stops detecting traffic orthogonal to the moving direction.

The `Dijkstra-starting-weight (DIJSW)` is the weight initially assigned to each edge when building the weighted graph, while `Dijkstra-high-weight (DIJHW)` and `Dijkstra-low-weight (DIJLW)` are respectively weight increment and decrement for an edge with associated information of low and high traffic. In a regular Manhattan street topology, to guarantee that empty streets are avoided, `DIJLW` must be at least three times `DIJSW`.[1] Moreover, to prevent an edge weight to become negative as a consequence of multiple notifications about high vehicular traffic along a certain street, the following equation should be satisfied: `DIJSW` $>$ `DIJHW` $\times street\_length/$`TID`, where $street\_length$ is the length of a street between two crossroads.

The `max-anchor-point (maxAP)` is the maximum number of APs that are included in a packet. This parameter is related with `maxTTL`: if `maxAP` is large with respect to `maxTTL`, some APs are computed without relying on traffic information. By contrast, a small `maxAP` could waste computation time because a vehicle refrains from using the information it owns to compute a longer path. Computing more APs allows greater accuracy in choosing a path; on the other hand, this implies higher overhead in both packet header and beacon traffic, as a consequence of the higher `maxTTL` needed.

4 Performance Analysis

STAR performance has been analyzed using the NS-2 simulation package [11] under different parameter settings, and has been compared with results achieved by GREEDY [10] approach without any recovery procedure, GPSR [8] approach, and SAR [9] algorithm. Simulations have been performed adopting a city mobility model, applied to a Manhattan street map formed by 5 horizontal streets and 5 vertical streets. Distances between adjacent streets are equal to 400 mt., thus characterizing a regular grid. The number of nodes is 250; the communication range is 250 mt. A small street length with respect to the communication range has been chosen to advantage GREEDY and GPSR, which have not been specifically designed for vehicular networks and do not involve mechanisms to deal with mobility along streets. Nodes move along the streets and can change their direction at crossroads; maximum vehicle speed is 50 Km/h. Simulated time is 200 sec. Results are averaged on 105 packets, exchanged between 5 source-destination pairs.

Simulations have been performed in three different scenarios: with all crossroads usable by vehicles, and with 4% and 8% of crossroads without vehicular traffic.

Initial measures have been performed varying 4 parameters, namely:

- `lowPR` threshold assuming values 0 or 1;
- traffic entry's `maxTTL` assuming values 10 or 20;
- `lowPE` threshold assuming values -2 or -4;
- `CROSS RANGE ACCEPT` parameter assuming values 10, 20 or 30.

[1] This way, a certain point can be reached avoiding an empty street by going around a building block via three streets.

Table 1. Parameter settings used in simulations

DIJSW	4	highPR	100
DIJLW	20	maxAP	5
DIJHW	-1	TID	150

Other parameters discussed in sec.3.3 are set to values reasonable in a real environment, as shown in Table 1. Detection of dense vehicular traffic conditions has been disabled because the mobility model prevents simulating queues of vehicles. Hence, `highPR` has a very high value while `highPE` does not even need to be defined. The performance indexes measured with simulations are:

- **percentage of delivered packets**;
- **percentage of lost packets:** packets can be lost because of collisions, or because the routing algorithm fails in getting them delivered to their destinations. This index only accounts for packets lost due to failures of the routing algorithm, with STAR performing only one path re-computation before deciding to drop the packet;
- **average beacon size:** large beacons increase the probability of collisions with both other beacons and data packets; hence, this index can explain other packet losses not due to routing failures.

Initial measures allowed us to tune the values of `lowPR`, `lowPE`, `maxTTL` and `CRA` in order to achieve the best performance. This is obtained with the scenario tending to minimize the beacon size, being prudent in both detecting and signaling absence of traffic (low `lowPR` and `lowPE` thresholds) and propagating the notification only in a restricted area (low `maxTTL`).

The probability of delivering packets increases with higher `CRA` value for increasing number of empty streets, because this increases the probability that the abnormal traffic condition is detected and advertised. Effective advertisement also reduces collisions: if a packet is appropriately routed by exploiting accurate information about the traffic distribution, then the probability of an alternative route computation is reduced, thus forcing the packet to follow a shorter route and decreasing packet collision probability.

The best scenario characterized through initial simulations uses `lowPR`=0, `maxTTL`=10, `lowPE`=-4 and `CRA`=20. The measures leading to this choice are reported in [12].

We compared the best scenario with the performance obtained by GREEDY, GPSR and SAR in the same conditions. It has to be noticed that a Manhattan street map simplifies route computation. Hence, map knowledge gives both SAR and STAR less advantages over the other two policies than expected, as there are not blind alleys or street forks. Moreover, SAR implementation does not involve any recovery procedure. Analyzing simulation results, it is worth to notice that so far a realistic mobility model is still missing. The model we used does not provide the possibility of simulating queues of vehicles, which also tend to be more dense around crossroads, thus having greater probability of packet collisions.

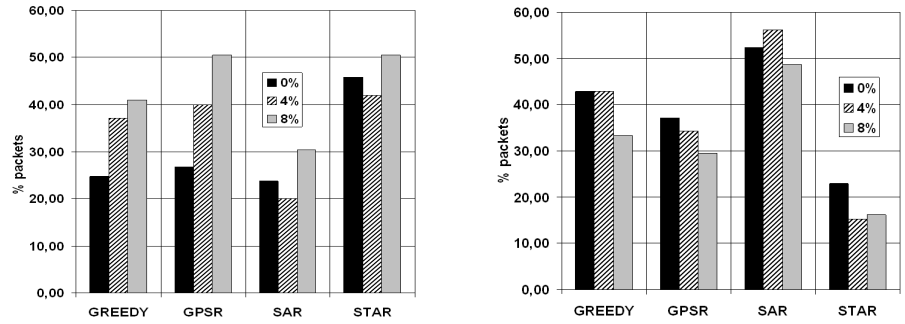


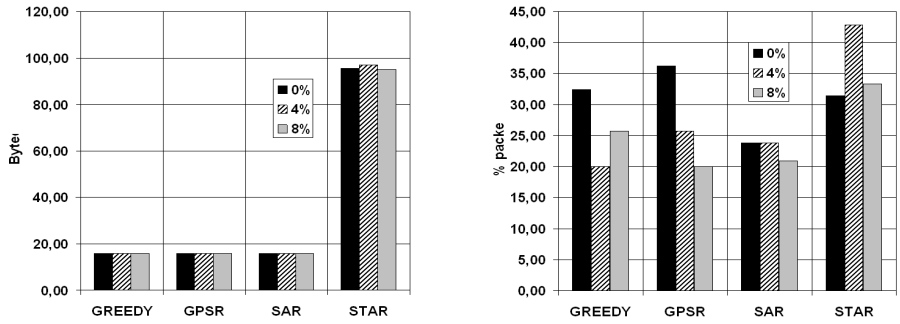
Fig. 2. Algorithm comparison: *(a)* percentage of delivered packets and *(b)* percentage of lost packets due to routing failure

Lack of queues and high collision probability worsen STAR performance with respect to a real environment. Moreover, the simulation environment does not allow to simulate radio signal attenuation due to obstacles: as a consequence, messages can be exchanged between two vehicles that actually would not be in communication range because of a building between them. This characteristic – together with streets short with respect to the communication range – advantages GREEDY and GPSR, while impacts to a lower extent on both SAR and STAR as they follow APs – and thus streets – to forward packets.

In fig.2*(a)*, the percentage of delivered packets is shown. STAR is comparable with GPSR, but it is advantaged by knowledge of street map. Both GREEDY and GPSR behave worse in the 0% scenario, because in that case vehicle density is lower than in the other scenarios as vehicles can be distributed all over the considered area. By fig.2*(b)* it can be noticed that STAR is far better than the other algorithms with respect to routing failure probability, thus confirming that STAR is effective in performing accurate routing decisions, preventing packets from reaching a local maximum. Indeed, the remaining packets not delivered to their destinations were lost because of collisions (fig.3*(b)*). On the other hand, although GPSR recovery procedure provides comparable guarantees of packet delivery in dense networks, the routes achieved with it are 1 to 2 hops longer than those computed with STAR.[2] SAR does not behave well because of both the lack of recovery mechanisms and the simplicity of the street map. If street length were higher, GREEDY and GPSR would drop many more packets because they are unable to find a longer route following streets to forward data to destination.

Distributing information about vehicular traffic drastically increases the size of network-layer beacons (fig.3*(a)*) with respect to the other considered algorithms. Because of larger beacons, collisions suffered by STAR overcome quite significantly those observed with other algorithms (fig.3*(b)*), except for scenario 0% where a lower average node density reduces collision probability.

[2] With path lengths of 7-8 hops on average.



(a) *(b)*

Fig. 3. Algorithm comparison: *(a)* average beacon size and *(b)* percentage of lost packets due to collisions

5 Conclusions and Future Works

In this paper we proposed a novel routing algorithm for vehicular ad hoc networks (STAR), and we measured its performance in comparison to other algorithms existing in the literature. STAR performs better than the other considered algorithms in spite of having an inaccurate mobility model that imposed disabling detection of high traffic conditions in simulations. However, STAR greatly suffers collisions. Parameters ruling traffic collection and information diffusion are the core of STAR: we are currently working on designing mechanisms for *dynamic adaptation of parameters*. A more realistic vehicular mobility model must also be developed, in order to ensure that more accurate results are obtained. Such a model would also allow to implement and evaluate the effectiveness of exchanging information about high vehicular traffic conditions. Anyway, this task is not trivial, as it involves correlating positions and speeds of different vehicles.

Acknowledgments

This work has been partially supported by the Italian Ministry of Education, University, and Research in the framework of the FIRB "Web-Minds" project.

We would like to thank Prof. Gian Paolo Rossi for useful discussions and his helpful comments in preparing the paper.

References

1. IETF MANET Working Group: *"Mobile Ad Hoc Networks"*. `http://www.ietf.org/html.charters/manet-charter.html`.
2. PaloWireless: *"Global Positioning System (GPS) Resource Center"*. `http://palowireless.com/gps/`.

3. Grossglauser, M., Vetterli, M.: *"Locating Nodes with EASE: Last Encounter Routing in Ad Hoc Networks through Mobility Diffusion"*. Proc. IEEE INFOCOM'03, Mar.2003.
4. Basagni, S., Chlamtac, I., Syrotiuk, V., Woodward, B.: *"A Distance Routing Effect Algorithm for Mobility (DREAM)"*. Proc. 4th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MOBICOM'98), pp. 76-84, 1998.
5. Li, J., Jannotti, J., De Couto, D.S.J., Karger, D.R., Morris, R.: *"A Scalable Location Service for Geographic Ad Hoc Routing"*. Proc. 6th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MOBICOM'00), pp. 120-130, 2000.
6. Stojmenovic, I.: *"Home Agent Based Location Update and Destination Search Schemes in Ad Hoc Wireless Networks"*. Technical Report TR-99-10, Computer Science, SITE, University of Ottawa, Sep. 1999.
7. Käsemann, M., Füssler, H., Hartenstein, H., Mauve, M.: *"A Reactive Location Service for Mobile Ad Hoc Networks"*. Technical Report TR-14-2002 , Department of Computer Science, University of Mannheim, Nov. 2002.
8. Karp, B., Kung, H.T.: *"Greedy Perimeter Stateless Routing for Wireless Networks"*. Proc. 6th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MOBICOM'00), Aug. 2000.
9. Tian, J., Han, L., Rothermel, K., Cseh, C.: *"Spatially Aware Packet Routing for Mobile Ad Hoc Inter-Vehicle Radio Networks"*. Proceedings IEEE 6th Intl. Conf. on Intelligent Transportation Systems (ITSC), Vol. 2, pp. 1546-1552, Oct. 2003.
10. Mauve, M., Widmer, J., Hartenstein, H.: *"A Survey on Position-Based Routing in Mobile Ad Hoc Networks"*. IEEE Network Magazine, Vol. 15, No. 6, pp. 30-39, Nov. 2001.
11. Fall, K., Varadhan, K.: *"The Network Simulator – NS-2"*. The VINT Project, `http://www.isi.edu/nsnam/ns/`.
12. Giudici, F., Pagani, E.: *"Spatial and Traffic-Aware Routing (STAR) for Vehicular Systems"*. Technical Report RT 07-05, Information and Communication Dept., Università degli Studi di Milano, Jun.2005.