Improving Routing Performance in High Mobility and High Density *ad hoc* Vehicular Networks

Rodolfo Oliveira[(1)(2)], Luis Bernardo[(1)(2)], Miguel Luis[(1)], Paulo Pinto[(1)(2)]

(1) FCT-UNL, Universidade Nova de Lisboa, Portugal

(2) UNINOVA, Monte da Caparica, Portugal

Abstract

In ad hoc networks the broadcast nature of the radio channel poses a unique challenge because the wireless links have time-varying characteristics in terms of link capacity and link-error probability. In mobile networks, particularly in vehicular ad hoc networks (VANETs), the topology is highly dynamic due to the movement of the nodes, hence an on-going session suffers frequent path breaks.

In this paper we present a method that uses the available knowledge about the network's topology to improve the routing protocol's performance through decreasing the probability of path breaks. We propose a scheme to identify long duration links in VANETs, which are preferentially used for routing. This scheme is easily integrated in the existent routing protocols. We describe how to integrate it in the Optimized Link-State Routing Protocol[1]. Finally, we evaluate the performance of our method with the original protocol. Simulation results show that our method exhibits better end-to-end path delay (almost one magnitude order lower) and packet delivery ratio (between 25% and 38% higher) than the original protocol. This observation is even more evident when the node's density increases.

Keywords: Topology Control, Routing Protocols, Vehicular *ad hoc* Networks.

[1]The source code of our proposal, entitled OLSR-FCT, was written for the network simulator ns-2.33 and is available to download at `http://tele1.dee.fct.unl.pt/people/rado/html/downloads .html`, allowing the community to evaluate their own scenarios and compare it with other protocols.

I. INTRODUCTION

Emerging vehicular networks are rapidly becoming a reality. Nowadays, several organizations are supporting standardization activities that will enable a variety of applications such as safety, traffic efficiency, and infotainment. Vehicular ad hoc networks (VANETS) share some common features with the traditional mobile ad hoc networks (MANETs), namely in terms of self-organization of the nodes. But they also differ in some issues: in VANETs the level of node's mobility is generally higher, the mobility is constrained by the roads and in terms of energy the nodes are not so constrained as in MANETs. Due to the fast change of the topology, VANETs demand for routing protocols focused on decreasing the number of path breaks.

The routing protocols that have been proposed for Mobile Ad Hoc Networks can be classified into three basic groups [1]: unicast topology-based, unicast position-based or group-based multicast and broadcast.

In topology-based protocols the nodes need to store routing tables or routes that depend on the topology. This class of protocols include the well known Ad hoc On-Demand Distance Vector Routing (AODV) [2], Optimized Link State Routing (OLSR) [3] and others (see [4], [5]). Traditionally these protocols were proposed for MANETs, where the nodes are assumed to have *moderate* mobility. This assumption allows these protocols the establish end-to-end paths that are valid for a reasonable amount of time and only occasionally need repairs. Therefore, these conditions are only valid in some vehicular scenarios, where the maximum speed of the nodes is strongly restricted. For high mobility scenarios, such as highways, the nodes exhibit unique characteristics [6] and the routing protocols used for MANETs do not perform well on VANETs [1]. For high mobility scenarios, topology-based protocols also pose other challenges related with the topology changes: usually they continuously maintain up-to-date routes for valid destinations and require periodic updates to reflect network topology changes. This requirement can lead to high bandwidth consumption, which can be alleviated by some optimization processes, such as the MultiPoint Relay (MPR) scheme used in [3].

In unicast position-based routing protocols [7], the nodes do not need to store any route or routing table to the destination. Instead, the nodes use the location of their neighbors and the location of the destination node to determine the neighbor that forwards the packet. Therefore, these routing schemes require information about the position of the nodes, which is a drawback when the positioning system fails (e.g. the GPS receivers can lose the signal inside tunnels).

In this work we approach the topology-based routing class, which does not requires positioning systems. We intend to use it in a high mobility scenario such as highways, to provide the deployment of comfort applications such as onboard games and video/music file sharing. Based on the fact that this class of protocols, namely those where the nodes store routing tables [2] [4] [3], use periodical broadcast of Hello messages to discover its neighborhood, we present a scheme to detect long duration links between vehicles. If properly used by the routing protocol, long-duration links are supposed to decrease the routing instability, decreasing the number of routing path breaks. The neighbors with which a node maintains long duration links are also identified and grouped. The groups are used to decrease the amount of broadcasts required to disseminate network topology changes.

Our approach is easily integrated in the existent routing protocols. We describe how to integrate it in the Optimized Link-State Routing Protocol [3] and we evaluate the performance of our method. Simulation results show that our method exhibits better end-to-end path delay (almost one magnitude order lower) and path availability for each destination (between 25% and 38% higher) when compared to the OLSR original protocol. This observation is even more evident when the node's density increases.

The rest of the paper is organized as follows. Section II motivates and describes the problem approached in this work. In Section III we describe how the long duration links are detected, and we introduce the algorithm that groups the neighbors which maintain long duration links. This section ends with an example of how our proposals can be incorporated in the Optimized Link-State Routing Protocol. Section IV presents the experimental results. Finally, some concluding remarks are given in section V.

II. Motivation and Problem Description

A. Motivation

The work presented in this paper is motivated by results obtained through simulations. We have simulated a VANET with 120 vehicles on a segment of a straight line highway with 3 lanes and 10 kms long. The simulation started with 30 vehicles moving on each side of the highway. During the simulation we launch more 30 vehicles on each side of the highway to maintain a constant density of nodes in the network. Each vehicle generates 0.5 packets per second and has, on average, 6 vehicles in its radio range[2]. The packets are randomly destined to the vehicles moving in the same way. The OLSR routing protocol was used.

In the first simulation, we evaluate the case when the multihop path only uses vehicles moving in the same way. Therefore, the radios of the nodes moving in one side of the highway were turned off. The simulated results, presented in table I, indicate that 68.8% of the packets were successfully delivered in an average time of 66.9ms. In the second simulation we repeated the first simulation setup, except that all vehicle's radios were turned on. Thus, the multihop paths can use the nodes moving in the opposite way. In this case, the routing performance diminishes almost 32% in terms of packet delivery ratio (from 68.8% in the first simulation to 47.1% in the second one). The same behavior is observed for the average end-to-end delay. It increases approximately 48% (from 66.9 ms to 99.1ms). These results indicate that the use of the vehicles in the opposite way can severely damage the routing performance in terms of both packet delivery ratio and end-to-end delay. These observations motivate this work, which aims to improve the routing performance for the scenario considered in the second simulation.

B. Problem Analysis

We start to consider the mobility scenario shown in Figure 1, where the vehicles 1 to 4 are moving at velocities $\vec{v}_1$ to $\vec{v}_4$, respectively. In this analysis we adopt the following assumptions: two nodes are d length unities far away from each other; the radio communication range of each node is expressed by r; a link is detected and subsequently sensed if $d \leq r$.

[2]Section IV gives more details about the simulated scenario.

TABLE I

OLSR RESULTS FOR MULTIHOP PATHS THAT ONLY USE NODES MOVING IN THE SAME WAY (SINGLE WAY) OR IN BOTH WAYS.

	single way	both ways
packet delivery ratio	68.8%	47.1%
average end-to-end delay	66.9 ms	99.1 ms

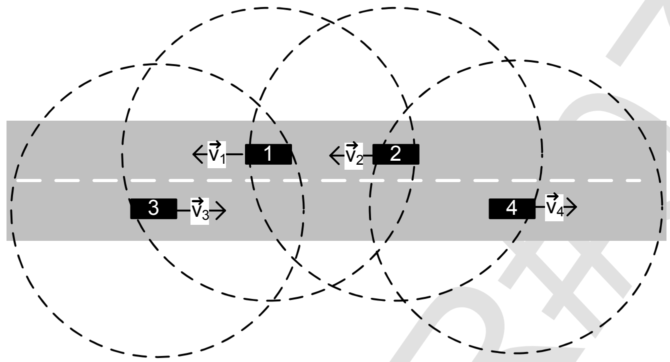

Fig. 1. Mobility scenario considered in the analysis.

Representing the velocity of the nodes n_a and n_b by the vectors $\vec{v}_a$ and $\vec{v}_b$ and being $d \leq r$, we differentiate two cases:

- $\vec{v}_a = \vec{v}_b$ - in this case the link between the nodes will remain active while this condition holds true (e.g.: $\vec{v}_1$ and $\vec{v}_2$ depicted in Figure 1);
- $\vec{v}_a \neq \vec{v}_b$ - this condition imposes that the link will be broken after some time (e.g.: $\vec{v}_1$ and $\vec{v}_3$ or $\vec{v}_2$ and $\vec{v}_4$ shown in Figure 1);

Representing $\vec{v}_a$ and $\vec{v}_b$ in polar coordinates (v_a, θ_a) and (v_b, θ_b), with $v_a, v_b \in [V_{min}, V_{max}]$ and $\theta_a, \theta_b \in \{0, \pi\}$, we represent the relative velocity of the nodes by

$$\begin{aligned} \vec{v_r} &= \vec{v_a} - \vec{v_b} \qquad (1) \\ &= (v_a cos(\theta_a) - v_b cos(\theta_b), v_a sen(\theta_a) - v_b sen(\theta_b)), \end{aligned}$$

and its absolute value is defined as

$$|\vec{v_r}| = g(v_a, v_b, \theta_a, \theta_b) = \sqrt{v_a^2 + v_b^2 - 2 v_a v_b cos(\theta_a - \theta_b)}. \qquad (2)$$

The relative velocity is a function that depends on the random variables v_a, v_b, θ_a, e θ_b, which are mutually independents. Considering a random variable V_r that expresses the relative velocity $V_r = g(v_a, v_b, \theta_a, \theta_b)$, the expected relative velocity value is defined as

$$E(v_r) = \int_{-\infty}^{+\infty} g(v_a, v_b, \theta_a, \theta_b) f(v_r)\, dv_r. \qquad (3)$$

As the random variables in (3) are independent, the condition $f(v_r) = f(v_a)f(v_b)f(\theta_a)f(\theta_b)$ is valid and $E(v_r)$ yields

$$E(v_r) = \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(v_a)f(v_b) f(\theta_a)f(\theta_b)\sqrt{v_a^2 + v_b^2 - 2v_a v_b cos(\theta_a - \theta_b)} \, d\theta_a \, d\theta_b \, dv_a \, dv_b. \tag{4}$$

Assuming that at instant t two nodes n_a and n_b form a link and, considering that the node n_a moves with velocity $\vec{v_r}$ relative to node n_b, the link will be considered broken if $|\vec{v_r}| > 0$ after some time. Assuming that nodes do not change their velocity between the interval $(t,\, t + \Delta t)$, which is a good approximation as $\Delta t \to 0$, the nodes will maintain a link active if during the interval $(t,\, t + \Delta t)$ the distance between the nodes never exceeds $2r$.

The probability of an existing link at time t remaining active in the time $t + \Delta t$ is related with the spacial intersection of the covered areas at instants t and $t + \Delta t$ (the space covered at both instants), which is represented by the shaded area in Figure 2.

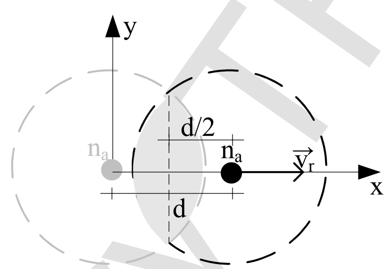

Fig. 2. Position of the node n_a in the time instants $t + \Delta t$ after moved d length units after the instant t.

Knowing that the radio covering circumference of the node n_a at instant t is given by

$$a_t = 2 \int_{-r}^{r} \sqrt{r^2 - x^2} \, dx, \tag{5}$$

the overlapped area in the instant $t + \Delta_t$ (represented by $a_{t+\Delta_t}(d)$ and illustrated by the shaded area in the Figure 2) is a function of the distance $d \geq 0$ travelled by the node n_1 with velocity v_r in the interval $(t, t + \Delta t)$, and is given by

$$a_{t+\Delta_t}(d) = \begin{cases} \pi r^2 - \int_{-d/2}^{d/2} \sqrt{r_2 - x_2} \, dx & 0 \leq d \leq 2r \\ 0, & d > 2r \end{cases} . \tag{6}$$

Now we consider the case when Hello messages are broadcasted every T_B seconds to discover and/or maintain an active link. The distance travelled by the node n_a relative to the node n_b during the period T_B is given by $E(v_r)T_B$. Therefore, the probability of the link remains active during $k\ T_B$ periods is given by

$$p_{link}(k) = \frac{a_{t+\Delta_t}(kE(v_r)T_B)}{\pi r^2}. \tag{7}$$

When the nodes n_a and n_b are moving in the same way, we have $f(\theta_a) = \delta(\theta_a), f(\theta_b) = \delta(\theta_b)$ or $f(\theta_a) = \delta(\theta_a - \pi), f(\theta_b) = \delta(\theta_b - \pi)$. In this case, the expected relative velocity value yields

$$E_{\text{same way}}(v_r) = \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} f(v_a) f(v_b) \sqrt{v_a^2 + v_b^2 - 2 v_a v_b} \, dv_a \, dv_b. \tag{8}$$

When the nodes n_a and n_b are moving in the opposite way, we have $f(\theta_a) = \delta(\theta_a), f(\theta_b) = \delta(\theta_b - \pi)$ or $f(\theta_a) = \delta(\theta_a - \pi), f(\theta_b) = \delta(\theta_b)$. In this case, the expected relative velocity value yields

$$E_{\text{opposite way}}(v_r) = \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} f(v_a) f(v_b) \sqrt{v_a^2 + v_b^2 + 2 v_a v_b} \, dv_a \, dv_b. \tag{9}$$

Because $E_{\text{opposite way}}(v_r) > E_{\text{same way}}(v_r)$ and $a_{t+\Delta(t)}(d)$ is a decreasing function in $0 \leq d \leq 2r$, the link between the nodes n_a and n_b has an higher probability (p_{link}) of remaining active when the nodes are moving in the same way. This conclusion should be adopted by the routing protocol: the multihop path created by the routing protocol should preferentially use links between nodes moving in the same way, because the lower probability of link breaks between these nodes will decrease the probability of path breaks.

III. Routing Improvements

A. Long Links Detection

Based on the description previously presented, this subsection introduces a solution to detect the links formed by two nodes moving in the same direction.

In topology-based routing algorithms the links between the nodes are discovered and maintained through periodical Hello packets exchange. The duration of the link is characterized by the number of Hello packets uninterruptedly received. We start to define the notion of node's **logical neighborhood**: the set of logical neighbors N_a is the set of 1-hop nodes from which the node n_a receives Hello packets.

In the time instant $t_i(n_y)$, when the node n_a firstly receives an Hello packet from it's neighbor node n_y, an unidirectional logical link is created between the nodes. The duration of the logical link can be quantified by its **stability** value: the stability $\eta(n_y)$ measures the duration of the logical link between the nodes n_a and n_y. $\eta(n_y)$ is computed by the node n_a at instant t by applying the following expression[3]

$$\eta(n_y) = 1 + (t - t_i(n_y)) \text{div} T_B. \tag{10}$$

Each node maintains its own table of logical neighbors to detect the links created in the same moving direction. Each line of the table represents one logical neighbor and contains information about n_a's neighbors address ($n_y \in N_a$), their stability value ($\eta(n_y)$), the time instant when the first Hello packet was received by n_a ($t_i(n_y)$) and a time interval ($T_O(n_y)$) during while the information described in the line is valid. Table II represents an hypothetic table of logical neighbors of node n_3 represented in the MANET of Figure 3.

TABLE II

TABLE OF LOGICAL NEIGHBORS OF NODE n_3 (ILUSTRATED IN FIGURE 3) AT THE INSTANT $t = 102.5s$ AND CONSIDERING $T_B = 1s$.

$N_3 = \{n_1, n_2, n_4\}$	$\eta(n_y), \forall n_y \in N_3$	$t_i(n_y)$	$T_O(n_y)$
n_1	65	37.2	k_1
n_2	30	72.3	k_2
n_4	2	100.1	k_4

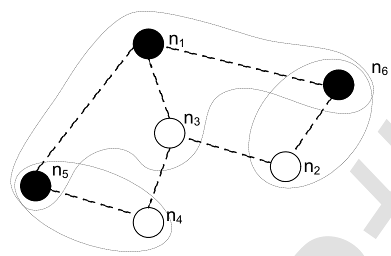

Fig. 3. MANET formed by 6 nodes.

The logical links between the nodes moving at the same direction are identified by each node through the observation of each link stability. A logical link is said to be stable if it lasts longer than a given k_{est} value:

$$\eta(n_y) \geq k_{est}. \tag{11}$$

By (6) and (7), a link created by two nodes moving in opposite directions presents a null probability of remaining active when $d > 2r$. Thus, for a link created by two nodes moving in opposite directions, the condition $p_{link}(k) = 0$ only holds when $k > 2r/(E_{\text{opposite way}}(v_r)T_B)$. Therefore, when the condition

$$k_{est} > \frac{2r}{E_{\text{opposite way}}(v_r)T_B} \tag{12}$$

holds, the stable links identified by the condition (11) represent the links maintained by the nodes moving in the same direction, because the links between vehicles moving in opposite directions never reach a stability $\eta(n_y)$ greater than the number k of T_B periods given by $2r/(E_{\text{opposite way}}(v_r)T_B)$. In the section IV we exemplify how to achieve $E_{\text{opposite way}}(v_r)$ and T_B for a real scenario.

B. Broadcast Leader Selection

The neighbors with which a node maintains stable links (stable neighbors) are more suitable to advertise network topology changes, because these nodes sense less link breaks. To decrease the amount of broadcasts needed to flood the network with the topology changes, we propose an algorithm that groups stable neighbors into 1-hop radius

[3]The symbol div is used for integer division.

groups. Each node selects a single Broadcast Group Leader (BGL). BGLs are preferentially used to broadcast the network topology changes. We denote $\xi(n_a)$ as the BGL node selected by the node n_a.

In the network depicted in Figure 3, the node n_4 selects n_5 has its own BGL, while node n_2 selects the node n_6. The remaining nodes n_1, n_3, n_5 and n_6 select the node n_1 as their BGL. Thus 3 different broadcast groups (BGs) are formed, represented by the sets $\{n_4, n_5\}$, $\{n_2, n_6\}$ and $\{n_1, n_3, n_5 n_6\}$. Now only the BGL nodes n_1, n_5 and n_6 are requested to broadcast in order to flood the entire network.

Considering a generic node n_a, and admitting that n_a knows the BGL nodes chosen by its neighbors ($\xi(n_y), \forall n_y \in N_a$ - which are transmitted in the Hello packet), n_a selects its own BGL by applying the Algorithm 1. The algorithm rules R1-R4 are summarized as follows:

```
Input   : N_a, η(n_y) (∀n_y ∈ N_a), ξ(n_y) (∀n_y ∈ N_a), t_i(n_y) (∀n_y ∈ N_a)
Output  : ξ(n_a)

η_max ⇐ return_max_η_from_beacon_table()
address ⇐ MAX_INT
ξ_aux ⇐ -1
transient_threshold ⇐ 1
if stable_node(n_a) then                                   /* R1 - if this node is stable */
    for each neighbor n_y ∈ N_a do
        insert_sorted(ξ(n_y), list_BGL)                    /* lower addresses in the head of the list */
    if (n_a is BGL) then
        insert_sorted(n_a, list_BGL)
    for each ξ_y ∈ list_BGL do                             /* ξ_y is removed from the head of the list */
        for each neighbor n_y ∈ N_a do
            if (n_y = ξ_y) and stable_node(n_y) then        /* R4 - select a neighbor that already is BGL */
                ξ_aux ⇐ n_y

        if (ξ_aux ≠ −1) then                               /* BGL already selected */
            break
        if (n_a = ξ_y) then                                /* R3 - auto-selection */
            ξ_aux ⇐ n_a
            break

    if (ξ_aux = -1) then                                   /* R2 - its neighbor becomes a new BGL */
        for each neighbor n_y ∈ N_a do
            if (η_max − η(n_y) − transient_threshold ≤ 0) and (n_y < address) then
                address ⇐ n_y
                ξ_aux ⇐ n_y



ξ(n_a) = ξ_aux
```

Algorithm 1: Algorithm used by the generic node n_a to select its Broadcast Group Leader $\xi(n_a)$.

R1 - when n_a is unstable (meaning that n_a does not have stable neighbors) it does not select any BGL;

R2 - when none of n_a's neighbors ($n_y \in N_a$) had previously selected a BGL, n_a selects the neighbor having the smaller address from the set of the neighbors with which n_a has the biggest stability value;

R3 - n_a selects itself as a BGL when n_a is already a BGL node (previously selected by a neighbor) and the neighbors's BGL have higher addresses than n_a;

R4 - when n_a is not selected BGL by its neighbors and there exists at least one neighbor n_y that is already a BGL, n_a selects the node n_y as its own BGL; ties are broken by choosing the smaller address neighbor;

The first BGL node selected in the network is justified by the application of the rule R2. The rule R4 is defined to merge several BGs. The rule R3 is also used to merge several BGs in the special case when n_a must selects itself as a BGL.

C. Integrating the topology information in OLSR Routing Protocol

This subsection exemplifies how the stable links and the Broadcast Groups are integrated in the Optimized Link-State Routing Protocol [3]. The OLSR routing protocol uses Multipoint relaying. Multipoint relaying is a technique to reduce the number of redundant re-transmissions while diffusing a broadcast message in the network [8]. Basically, a node n_a chooses a set of Multipoint Relay nodes (MPRs). MPRs are chosen as the minimum set of n_a's 1-hop neighbors that cover all its neighbors 2-hops away. Thus MPR nodes guarantees 2-hops full coverage.

We use the BGL nodes in the MPR's selection algorithm, proposing a new algorithm to compute the MPRs (Algorithm 2). For a given node (n_a), the algorithm needs to know about its neighbors ($n_y \in N_a$), their stability ($\eta(n_y)$), their BGLs ($\xi(n_y)$) and the required coverage. OLSR uses the smallest set of MPRs that guarantees the full coverage 2-hops away. Because our algorithm does not guarantees the smallest set of MPRs to cover the neighbors 2-hops away, it may introduce undesired control traffic overhead. This is the main reason to design our algorithm with an input parameter named `required_coverage`, that allows us to characterize our algorithm's behavior for different amounts of coverage.

Input : N_a, $\{\eta(n_y), \xi(n_y), t_i(n_y)\} \forall n_y \in N_a$, required_coverage

Output : MPR(n_a)

```
MPR(n_a) ⇐ φ
coverage ⇐ 0
for each_neighbor n_y ∈ N_a do
    if η(n_y) ≥ k_est and n_y is BGL then          /* R1 - if n_y is stable and BGL */
        MPR(n_a) ⇐ MPR(n_a) ∪ n_y
        update(coverage)
        remove_from_list(n_y, N_a)

N_ord = sort_by_descendent_η(N_a)
while coverage < required_coverage do
    remove_from_list(n_y, N_ord)
    MPR(n_a) ⇐ MPR(n_a) ∪ n_y
    update(coverage)
```

Algorithm 2: New algorithm to select OLSR's MPR nodes used by the OLSR.

The algorithm is summarized as follows. 1-hop neighbors of n_a that have been selected BGLs and have stable links with n_a are selected MPRs (lines 3 to 8). If the number of MPRs are not guaranteing the required coverage 2-hops away (line 10), the algorithm selects the MPR nodes with which n_a has the highest stability values.

Regarding the routing table's computation, the OLSR uses a field named Willingness that specifies the willingness of a node to carry and forward traffic for other nodes. Only 1-hop neighbors with Willingness different of `WILL_NEVER` are used to forward packets. In our proposal, we use the OLSR's Willingness mechanism to prohibit n_a's unstable neighbors from forwarding packets. We assign the willingness of n_a's 1-hop neighbors that do not meet the condition stated in (12) to `WILL_NEVER`.

IV. Performance Evaluation

A. Mobility Model

The VANETs simulated in our evaluation scenario were obtained using the Trans tool [9], which integrates the SUMO traffic simulator [10]. We have simulated a segment of a straight line highway with 3 lanes in each direction. The simulations start with the vehicles moving in both sides of the highway. During the simulation we launch more vehicles to maintain a constant density of nodes in the network. The highway segment is 10 kms long, which limits the minimum number of the network hops to 10. We defined three different classes of vehicles, which are described in the table III. 60% of the vehicles belong to the class_1, which represents medium size cars. The vehicles of class_2 represents 25% of the highway traffic. Finally we define 15% of vehicles of class_3, which represents long sized vehicles such as trucks. Regarding the vehicle's density, we defined 3 different scenarios, described in the table III. All vehicles are assumed to have a radio range of 1000 meters. The scenario Scen_6 was the same used to obtain the motivation results presented in the table I.

TABLE III

Classes of traffic considered in the simulations.

	vehicle's length (m)	V_{MAX} (m/s)	acceleration (m/s^2)	deceleration (m/s^2)	%
class_1	4	27.8	3.6	3.6	60
class_2	5	26.0	2.5	3.0	25
class_3	8	20	1.5	2.0	15

B. Simulation Description

The simulations compare the performance of our improvements with the OLSR routing protocol. We used the simulator ns-2 [11] configured with the standard IEEE 802.11[4] and the OLSR protocol implementation available

[4] 11 Mbps and 2 Mbps were used to transmit unicast and broadcast traffic, respectively.

TABLE IV

VEHICLE'S DENSITIES CONSIDERED IN THE SIMULATIONS.

	number of vehicles	average number of neighbors	simulation time (s)
Scen_6	120	6.0	830
Scen_8	160	8.0	915
Scen_{10}	200	10.0	886

from [12]. Our proposals were integrated in the OLSR implementation, being the modified protocol designated OLSR-FCT.

Vehicles moving in the same left-to-right direction generate 0.5 packets per second, which are randomly destined to the nodes in the same direction. The vehicles moving from right-to-left do not generate packets but are able to forward them. The number of packets generated on each density scenario were 7388, 10102 and 13220[5] (from Scen_6 to Scen_{10}, respectively). We simulated 6 different situations: 3 density scenarios previously described using the OLSR protocol and the same scenarios using our OLSR-FCT protocol.

To parameterize k_{est} we assumed a rough approximation for the three classes of traffic, as being normally distributed with and average V_{MAX} and 0.5m/s of standard deviation. For this case the relative velocity between two vehicles moving in opposite directions yields $E_{\text{opposite way}}(v_r) = 52.36$ m/s, and the relative velocity between two vehicles moving in same direction ($E_{\text{same way}}(v_r)$) yields approximately 2m/s. Choosing the Hello transmission frequency (T_B) to 1Hz, the minimum k_{est} value that guarantees unstable links formed by opposite moving vehicles is 38.19 (given by (12) and considering $r = 1000$m). Therefore, we chosen k_{est} = 50 to have a security margin of approximately 12 T_B periods. This margin avoids the vehicles moving at lower speeds to be erroneously declared as moving in the same direction. This parameterizations indicates that the links of the vehicles moving in the opposite direction never last longer than 50s. Moreover, when the links of the vehicles moving in the same direction last for 50s, their probability of still being active is 96.8% ($p_{link}(50) = a_{t+\Delta_t}(50 \times 2 \times 1)/(\pi \times 1000^2)$).

C. Experimental Results

The OLSR protocol was firstly simulated for the 3 scenarios described in table IV. The path delivery ratio and the average end-to-end delay were obtained within 95% of confidence interval and are shown in table V. The results indicate that the packet delivery ratio is approximately 50% in the 3 scenarios and the end-to-end increases with the vehicle's density.

The results obtained with our OLSR-FCT proposal are shown in the Figures 4 and 5. The OLSR-FCT was tested for different values of `coverage` (45%, 60%, 85% and 100%, and the `coverage` means the same as defined in

[5]Note that a vehicle only generates packets when it is moving.

TABLE V

OLSR EXPERIMENTAL RESULTS.

	packet delivery ratio (%)	end-to-end delay (ms)
$Scen_6$	47.13%	99.10
$Scen_8$	53.94%	221.37
$Scen_{10}$	52.75%	301.59

the Algorithm2). The results obtained with the OLSR-FCT protocol present a better performance when full coverage is used (100%). For full coverage the number of MPRs increase and, consequently, more topology traffic overhead is generated. This fact increases the end-to-end delay (Figure 5 shows higher average end-to-end delays when the coverage is 100%). The results plotted in Figure 4 show that the OLSR-FCT protocol exhibits better packet delivery ratio for higher node densities.

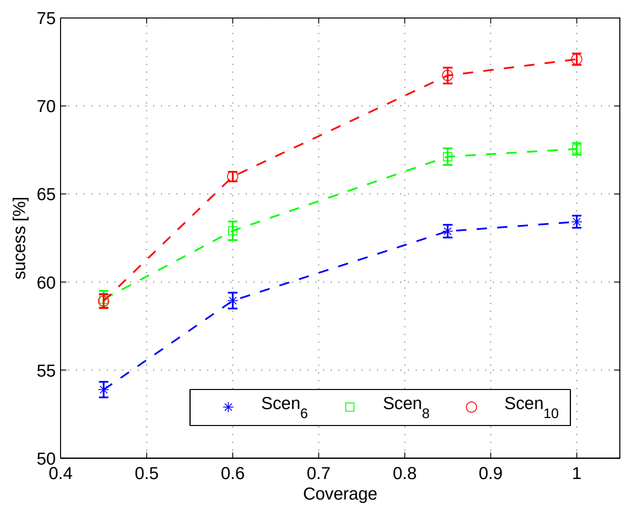

Fig. 4. Average packet delivery ratio (95% confidence intervals represented by vertical bars).

Comparing the results presented in the motivation (obtained with the original OLSR protocol) we conclude that the OLSR-FCT protocol improves the packet delivery ratio (from 47.1% to 66.9%) and almost achieves the same ratio as in the case when the radios of the vehicles in one direction were turned off (68.8%). In terms of end-to-end delay, the OLSR-FCT decreases the delay from 99.1ms to 42.2ms, being even shorter than the delay measured for the case when the radios of the vehicles in one direction were turned off (66.9ms). This fact is due to the use of long-duration links and control traffic decrease implemented by the OLSR-FCT protocol.

Finally, we present the gains obtained with OLSR-FCT in the table VI.

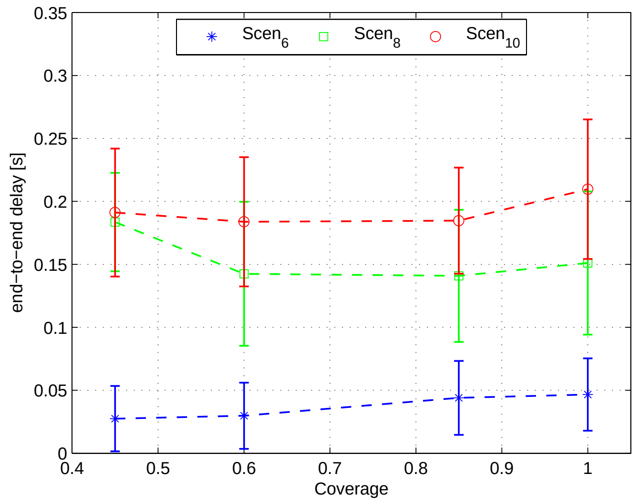

Fig. 5. Average end-to-end delay (95% confidence intervals represented by vertical bars).

TABLE VI
OLSR-FCT EXPERIMENTAL GAINS WHEN 100% OF COVERAGE IS CONSIDERED.

	packet delivery ratio gain (%)	end-to-end delay gain (%)
$Scen_6$	33.2%	52.94
$Scen_8$	26.19%	31.76
$Scen_{10}$	37.72%	30.48

V. Conclusions

In this paper we present a method that uses the available knowledge about the network's topology to improve the routing protocol's performance through decreasing the probability of path breaks.

We integrate our improvements in the OLSR routing protocol. The performance results explicitly confirms that our proposal outperforms the original protocol, recommending its use for high mobility and high density scenarios.

References

[1] Jasmine Chennikara-Varghese, Wai Chen, Onur Altintas, and Shengwei Cai. Survey of routing protocols for inter-vehicle communications. In *Mobile and Ubiquitous Systems: Networking and Services, 2006 Third Annual International Conference on*, pages 1–5, July 2006.

[2] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, 25-26 Feb 1999.

[3] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol. *IETF Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-10.txt*, 2003.

[4] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.

[5] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[6] J.J. Blum, A. Eskandarian, and L.J. Hoffman. Challenges of intervehicle ad hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):347–351, Dec. 2004.

[7] M. Mauve, A. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6):30–39, Nov/Dec 2001.

[8] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. *Hawaii International Conference on System Sciences*, 9:298, 2002.

[9] TraNS. - open source tool for realistic simulations of VANET applications . Software Package retrieved from http://trans.epfl.ch/, 2009.

[10] Centre for Applied Informatics (ZAIK) and Institute of Transport Research at the German Aerospace Centre. SUMO - Simulation of Urban Mobility. Software Package retrieved from http://sumo.sourceforge.net, 2009.

[11] Information Sciences Institute. NS-2 network simulator (version 2.31). Software Package retrieved from http://www.isi.edu/nsnam/ns/, 2007.

[12] Francisco J. Ros. *UM-OLSR. Software Package retrieved from http://masimum.inf.um.es/um-olsr/html/, 2009.*