Accepted Manuscript

Kinetic Mobility Management Applied to Vehicular Ad Hoc Network Protocols

Jérôme Härri, Christian Bonnet, Fethi Filali

PII: S0140-3664(08)00044-3
DOI: 10.1016/j.comcom.2008.01.060
Reference: COMCOM 3581

To appear in: *Computer Communications*

Received Date: 14 July 2007
Revised Date: 14 January 2008
Accepted Date: 14 January 2008

Please cite this article as: J. Härri, C. Bonnet, F. Filali, Kinetic Mobility Management Applied to Vehicular Ad Hoc Network Protocols, *Computer Communications* (2008), doi: 10.1016/j.comcom.2008.01.060

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Kinetic Mobility Management Applied to Vehicular Ad Hoc Network Protocols

Jérôme Härri , Christian Bonnet , Fethi Filali

Institut Eurécom
Département Communications Mobiles
B.P. 193, 06904 Sophia-Antipolis – France

Abstract

Vehicular Ad Hoc Networks (VANETs) are a particular category of mobile ad hoc networks (MANETs) characterized by a high mobility and a reduced connectivity. In order to develop protocols for vehicular networks, the community may either create VANET specific approaches, or adapt already existing protocols to VANET. While the former may provide efficient specialized solutions, the latter offers an increased interoperability and universality, which is a key issue for industrial partners involved in the deployment of VANET and Intelligent Transportation Technologies (ITS). An important aspect in the porting of ad hoc networks solutions to VANET and ITS is an efficient management of vehicular mobility.

Mobility Management is a principle aimed at updating network routes or structures in order to keep them coherent with mobile topologies. Mobility management may be proactive or reactive, depending if the updates are triggered with or without topology changes, or if and only if a change in the topology effectively requires to update the structure. Failure to develop efficient mobility management heuristics leads to a waste of network resources and suboptimal routes or structures. The optimal solution is obviously the reactive mobility management, as updates are optimally triggered only when necessary. However, due to its complexity, the reactive mobility management has not attracted as much attention as its proactive counterpart.

In this paper, we introduce a location-aware framework, called Kinetic Graphs, that may be followed by ad hoc protocols in order to implement a reactive mobility management. The Kinetic Graph framework is able to capture the dynamics of mobile structures, and is composed of four steps: (i) a representation of the trajectories, (ii) a common message format for the posting of those trajectories, (iii) a time varying weight for building the kinetic structures, (iv) an aperiodic neighborhood maintenance. We eventually provide a example of a successful application of this framework to broadcasting and routing in VANET.

Key words: Mobility Management, Kinetic graphs, mobility predictions, broadcasting, routing, ITS, vehicular networks.

Email addresses: haerri@eurecom.fr (Jérôme Härri), bonnet@eurecom.fr

ACCEPTED MANUSCRIPT

1 Introduction

Vehicular communication is regarded as a key technology for improving road safety and comfort through Intelligent Transportation Systems (ITS). The growing interest towards the possible applications of wireless technologies to the vehicular environment has recently led consortia (US VII [1], EU C2C-CC [2]) and standardization bodies (IEEE [3]) to develop technologies and protocols for transmission of data between vehicles and between vehicles and road infrastructures. A network without any centralized coordinator and where communicating nodes compose cars or road infrastructures is called a Vehicular Ad Hoc Network (VANET). VANET is therefore a generalization of MANET for extremely mobile topologies. Due to the particular mobility and connectivity, standard protocols for MANET have been mostly criticized due to the latency and overhead required when used for vehicular networks. It has been notably observed that the OLSR routing protocol based on the Multipoint Relaying structure was not adapted to highly mobile networks such as vehicular networks, and more generally that proactive routing protocols consumed a significantly large energy and network resource dedicated to the maintenance of their routing tables. The community working in vehicular communication therefore started to develop solutions specific to VANETs, geographic routing such as Greedy Perimeter Stateless Routing (GPSR) or Last Encounter Routing (LER) for instance, or more generally opportunistic routing.

Taking a different look, we can see that the limitation of standard routing and broadcasting protocols for highly dynamic networks comes from the lack of an efficient management of nodes mobility and not the protocols themselves. Indeed, routing data is built and optimized based on topology information gathered by periodic beacon messages. Beside the significant overhead of periodically sending topology information, this critical procedure is also limited by its stability with respect to the latency required to update routing data and the validity of topology information. If nodes are moving too fast, the time needed to update routing tables might actually exceed the duration of the links composing the routing paths for instance.

One solution in order to improve mobility management is to increase the time interval during which the topology is assumed known and does not need updates. For that matter, mobility predictions could be used in order to avoid dead links by predicting alternate connectivity solutions. As long as topology information is correctly predicted, a maintenance is not required. Accordingly, the maintenance is optimized by updating routing data if and only if an unpredicted new topology information truly affects routing. Thanks to this enhanced mobility management that we categorize as *kinetic*, the use of standard MANET routing protocols, such as DYMO, MPR or OLSR, could be envisioned again for VANET and ITS. That is also a significant argument for industries and standardization bodies working

(Christian Bonnet), `filali@eurecom.fr` (Fethi Filali).

ACCEPTED MANUSCRIPT

in ITS and vehicular communication, as it could ease the interoperability between vehicular networks, and fixed networks or MANETs, where these protocols lead.

In this paper, we propose a Kinetic Mobility Management solution based on mobility predictions for optimizing standard MANET protocols for VANETs or ITS. We define a location-aware framework called the *Kinetic Graph* that may be followed by ad hoc protocols in order to reach a kinetic mobility management. We first provide a general description of how the trajectories are modeled, how structures are initially built and finally, how they are maintained. We emphasis that our objective is to suppress the periodic beaconing process widely used by almost all ad hoc protocols in order to adapt to mobility, and also to increase the time interval during which the mobile topology is correctly anticipated. Then, we discuss two different kinetic link weights that could be easily adapted in most of the protocols for VANETs. Finally, we provide a successful application of the Kinetic Graph approach to broadcasting in VANETs. We also would like to emphasize this approach may also be applied to VANET and ITS specific protocols, as improving mobility management is also important for geographic routing for instance.

The rest of this paper is organized as follows. In Section 2, we define a novel terminology for mobility management and describe the challenges of this terminology in VANETs. Section 3 formally introduces the Kinetic Graphs and covers the four steps of its framework, while Section 4 provides an application example of the Kinetic Graph framework to broadcasting in VANETs. We finally provide some related works in the field of mobiltiy management in Section 5 and conclude in Section 6 with some insights on future orientations of kinetic mobility management.

2 Reactive Mobility Management

In this section, we define a novel and optimal concept for mobility management called *Reactive Mobility Management*, which is based on mobility predictions and aims at optimally updating a structure when and only when required, regardless of any topology change. First, we define the concept, then address the challenges facing this concept, and finally analyze the expected performance of this concept.

2.1 Definition of Concept

The MANET routing community classified routing protocols mostly in two classes *Proactive* and *Reactive*, depending if a route is created when there is data traffic to transmit, or if all routes are proactively created independently of the data traffic.

ACCEPTED MANUSCRIPT

As mobility was not considered during the design of most routing protocols, it is only handled by a periodic maintenance process in order to detect any topological change and update the routes. This process is suboptimal and waste network resources, as the process is run even if nothing needs to be changed. Taking a similar vocabulary, but considering mobility instead of routing, routing protocols may therefore be considered to be using a proactive mobility management defined as follows:

Definition 1 (Proactive Mobility Management) *A Proactive Mobility Management protocol proactively triggers a maintenance process with or without topology changes. Moreover, as proactive protocols do not have any vision of future topologies, the process is usually repeated periodically.*

According to this definition, a mobility proactive protocol may only adapt the routes based on past topologies, as nodes already moved when the process starts. Moreover, important network resources are wasted as the maintenance is triggered with or without changes in the topology that effectively requires to update the routes.

In opposite, a better choice would be to only start a maintenance process reactively to a change that requires an update. We therefore have the following definition for the reactive mobility management approach:

Definition 2 (Reactive Mobility Management) *A Reactive Mobility Management protocol tries to anticipate all topological changes using mobility prediction techniques and only starts a maintenance process reactively to a missed prediction. Moreover, as all nodes are assumed to run the same prediction schema, the node that wrongly predicted its own behavior is responsible for triggering the maintenance process.*

According to this definition, a reactive mobility management protocol will not act as long as the topology evolves as predicted, and only react if a change in the topology effectively affects the routes. This is an optimal mobility management, as network resource is not wasted for any unnecessary maintenance.

We therefore use a similar terminology but in a different application. For data traffic, proactive protocols open all routes with or without traffic, while reactive protocols open routes if and only if there is traffic to send on that route. With respect to mobility, a proactive mobility management protocol triggers a maintenance duty with or without change in the network topology, while the reactive one triggers a maintenance duty if and only if there is an unanticipated topological change that effectively impacts the structures.

ACCEPTED MANUSCRIPT

2.2 Challenges of Predicting Mobility

As the potential gain from the reactive mobility management concept comes from its capability to correctly predict a future topology, the challenge is therefore to design efficient mobility prediction techniques. The performance of a mobility prediction schema may be quantified by evaluating its prediction error. Yet, a prediction schema is a complex process, and potential prediction errors may come from different factors. Similarly to the analysis of mobility models, we propose to decompose a mobility prediction schema into functional blocks, then analyze the error created by each block.

The objective of a mobility prediction schema is to predict real movement patterns or traces. Depending on the approach, a mobility model generates traces, or the model is extracted from traces. In both cases, the prediction schema is based on a mobility model and its patterns. Therefore, the first step in predicting mobility is modeling this mobility. It is therefore straightforward to see that the first source of potential prediction error comes from the realism of a mobility model with respect to the real movement patterns. If the traces are not accurately modeled by a mobility model, this error will not be able to be corrected by even the most complex prediction models.

Definition 3 (Realism) *The realism is the depiction of a feature as it appears in life, without error, interpretation or embellishment. A mobility model, thus its functional blocks, should therefore be as realistic as possible.*

The realism of a mobility model may be evaluated by comparing its synthetic traces with real traces obtained through a measurement campaign. If the difference lies within an acceptable gab, one say that the mobility model has been validated, and thus has a small realism error. If not, the realism error may be significant. For example, modeling vehicular motions with the random waypoint model generates a significant realism error. As this process of validating a mobility model requires significant financial and manpower resources, another solution is to compare a mobility model with another model that has already been validated. That is the solution chosen by VanetMobiSim [4], a realistic and validated vehicular mobility model that we will use in this paper. Although each application has its realistic mobility model, such as pedestrian or or vehicular mobility, we targeted the latter as vehicular mobility patterns show non-uniform distributions of cars and velocity coming from a strongly restricted mobility helping to reduce the similarity error. Moreover, the concept of trajectory may be easily seen in vehicular motions.

Macroscopically speaking, a mobility model may be decomposed into two function blocks:

- **Trajectory Modeling–** It is defined as the probable course of a node in a mobile system

ACCEPTED MANUSCRIPT

- **External Configuration–** It is composed of random configuration parameters, as well as external influences on the trajectory modeling, such as the impact of traffic accident on traffic management

A prediction model must act on both blocks, and its errors will be evaluated with respect to its ability to follow each block. Therefore, a prediction model may also be separated into the following blocks:

- **Kinematic Model–** The kinematic model aims at reproducing the trajectory modeled by the mobility model as adequately as possible. As a prediction model may only have access to a limited information on the real trajectories, a variance exists between the modeled and the predicted trajectories creating an adequacy error.
- **Kinematic Hypothesis–** Since the complexity of most of the real motion patterns exceeds the ability to develop a kinematic model, the problem is usually relaxed using hypothesis on the kinematics patterns. For example, a model could assume a fixed velocity between two successive trajectory changes. If the hypothesis is invalid, then the adequacy error is further increased.
- **Prediction Criterion–** The prediction criterion is the mechanism that detects a change in the motion patterns according to the prediction model. When the criterion is invalidated, the prediction model creates a wrong prediction and thus needs to be updated. The criterion therefore controls the relative predictability of the motion patterns modeled by the prediction schema. Examples of prediction criteria are distance between two nodes, or constant speed.

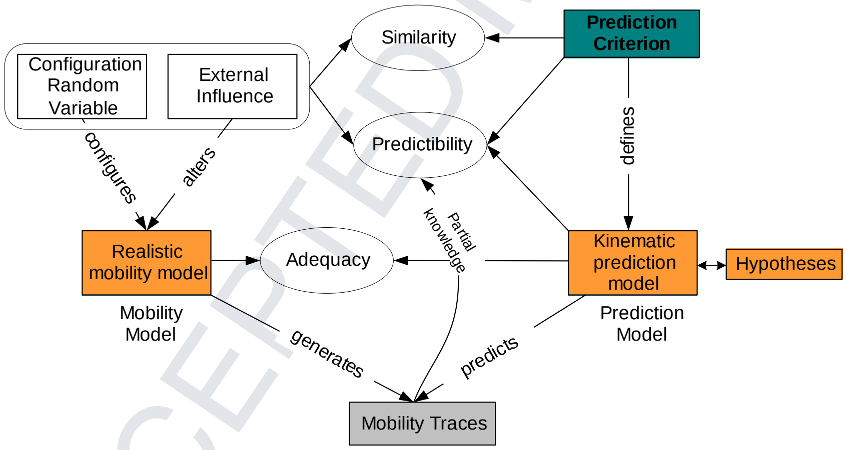


Figure 1. Illustration of the relationship between Realistic Mobility and Prediction Models

Figure 1 illustrates the interactions between the mobility modeling and prediction blocks. The accuracy of the interaction between the different blocks is represented by prediction errors, which we classified in four categories.

Definition 4 (Adequacy) *The adequacy reflects the similarity between the motion models used by the kinematic model and the modeled trajectories. If the two models are identical, or yield to identical results, one say that the two models are adequate.*

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

The hypothesis plays a crucial role in the adequacy and should thus be wisely chosen.

Definition 5 (Predictability) *The predictability is a time interval between which the prediction criterion remains valid or within an acceptable error ϵ.*

Definition 6 (Similarity) *The similarity reflects the relative variance of the modeled trajectories between successive criteria. Independently to the adequacy, it reflects the extends of the error if predictability is miscalculated or not available.*

The adequacy is obtained by the comparison between the synthetic kinematic models employed by the prediction schema and the mobility model. The adequacy is maximized when the two models are identical, and it is minimized when they totally diverge. Usually, the objective is to develop kinematic models that fit best to real mobility patterns. For example, using a kinetic first order motion $x = v \cdot t + x_0$ to model vehicular mobility is highly inadequate and will lead to a strong divergence of the prediction model. In opposite, using such linear model on the Random Waypoint Model leads to a perfect adequacy.

As the time during which the hypothesis used by the kinematic prediction model remains valid also controls the time interval during which the criterion remains valid, the predictability depends on the analysis of the stability of the hypothesis. However, this study cannot be obtained in real-time, but only *a posteriori*, as the prediction model only has a partial access to the mobility model's parameters. For instance, a car may be able to transmit its position and velocity but not its destination, as it might be unknown, subject to external factors, or simply subject to privacy protections. Accordingly, an average predictability must be learned based on the history of previous movement patterns, or statistically obtained if the motion patterns are modeled by an analytical model.

The similarity is obtained by measuring the variance between successive values of the criterion. The similarity is minimized when no correlations exist between past and future values, while it is maximized when the future value may be fully extracted based on past ones. For example, vehicular mobility may have complex patterns but benefit from a large similarity.

Now that we defined the different relationships between the various functional blocks in Figure 1, we are ready to define the prediction error generated by a prediction schema. As this error depends on the criterion to be predicted, we use the general term "Criterion Prediction Error", and define it with respect to the mobility patterns.

Definition 7 (Criterion Prediction Error) *It represents the order of magnitude between the true and the predicted criteria. The objective is to minimize this error by either changing the sensitivity of the criterion with respect to mobility prediction errors, or improve the parameters controlling this error. The criterion error is* *defined as*

$$O(\gamma + \tau + \xi \cdot \upsilon)$$

where

- γ*: represents the realism error*
- τ*: represents the adequacy error*
- ξ*: represents the predictability error*
- υ*: represents the similarity error*

We summarize the process of predicting mobility and illustrate the criterion prediction error in Figure 2. Assuming the real trajectory followed by a node started at time $t = 0$, the first step is to model that trajectory with a mobility model. If the modeled trajectory is not similar to the "real" trajectory, we create a realism error γ. The next step is to predict the modeled trajectory with a kinetic prediction model. Once again, if the predicted and the modeled trajectories are not identical, we generate an adequacy error τ. The predicted trajectory is considered valid during the average predictability interval λ. If the criterion changes before the end of this interval, a predictability error ξ is generated, which is illustrated at time $t = 15$ on Figure 2. Yet, the extends of this error also depends on the similarity. Indeed, if two successive criteria are close, the predictability error is minimized. In the contrary, the error will be maximized. This is illustrated by the case (1) or (2) on Figure 2.

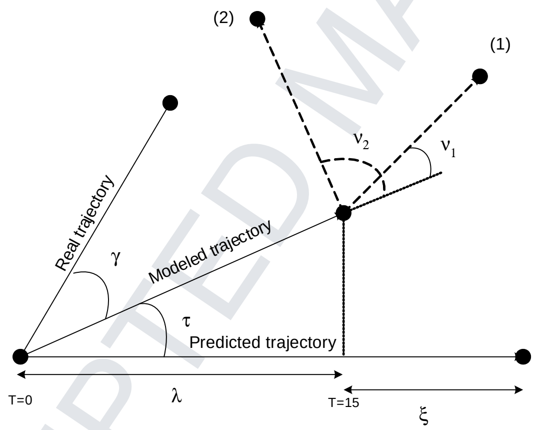


Figure 2. Illustration of criterion prediction error

2.3 Performance Evaluation

Previously, we introduced the novel concept of reactive mobility management, and also provided a definition of the errors generated by a prediction model when used in conjunction with the reactive mobility management. Even though the performance of a prediction model is provided by the analysis of the prediction error, the performance of the reactive mobility management approach is actually controlled by the predictability interval λ, defined as the time interval between two successive

ACCEPTED MANUSCRIPT

valid criteria. Indeed, as the reactive approach starts a maintenance process roughly at the instantaneous predictability interval (i.e. when the criterion is invalidated), a frequent maintenance is required if λ is short, while a seldom maintenance may be reached if λ is long.

When the predictability interval is significantly reduced or too small to be efficiently used, the reactive maintenance falls into a degenerated case which is equivalent to the proactive maintenance, where the maintenance is periodically performed. Accordingly, the performance of a reactive mobility management tends to a proactive mobility management.

Two parameters are therefore required to evaluate the performance of the reactive mobility management: the *prediction error* and the *prediction performance*. By comparing them for various strategies, we may evaluate what is the best choice for a mobility protocol:

- **Proactive Strategy:** The node periodically starts the maintenance process at each time interval $\kappa \leq \lambda$.
  · *Prediction Error:* $O(\gamma + \tau + \kappa \cdot \upsilon)$.
  · *Prediction Performance:* $O(\kappa)$.
- **Adaptive Strategy:** The node which generated the predictions corrects them at the end of the average predictability interval.
  · *Prediction Error:* $O(\gamma + \tau + \xi \cdot \upsilon)$.
  · *Prediction Performance:* $O(\xi + \lambda)$.
- **Reactive Strategy:** The node which criterion changed immediately notifies the neighborhood. The predicted trajectory is therefore corrected at the exact predictability interval λ.
  · *Prediction Error:* $O(\gamma + \tau)$.
  · *Prediction Performance:* $O(\lambda)$.

As it may be seen, the major benefit of the reactive approach is that the predictions are updated roughly at the same time as the mobility parameters, canceling the predictability and the similarity errors. This yet comes of a performance depending on the exact predictability interval λ. The challenge is therefore to jointly address the adequacy and realism together with predictability.

3 The Kinetic Graph Framework

In the previous section, we defined the reactive mobility management, as an optimal management strategy for mobile networks. The objective of this section is to define a framework indicating the guidelines to follow in order to implement a reactive mobility management protocol. Most of the protocols in MANETs are based on graph theory. Although this field created efficient algorithms for ad hoc networks,

ACCEPTED MANUSCRIPT

its major drawback remains its limitation to quasi-static networks and to proactive mobility management. Graph Theory is also a good candidate for the implementation of reactive mobility management schemes, as any MANET protocol based on graph theory could benefit from the proposed framework.

In this section, we therefore introduce the **Kinetic Graph Framework** to adapt any graph algorithm to reactive mobility management. The framework consists of four steps: (i) a representation of the trajectories, (ii) a common message format for the posting of those trajectories, (iii) a time varying weight for building the kinetic graphs, (iv) an aperiodic neighborhood maintenance. We also provide two examples of possible time-varying weights. The framework does not specifically target a particular area of MANETs, but we will provide later in this paper an application example of its use in the case of broadcasting in Vehicular Ad Hoc Networks.

3.1 Preliminary Concept Definitions

Before moving forward, we provide some necessary preliminary definitions related to graph theory. In *static* graph theory, the following definitions are usually used:

- **Link Weight –** It is a value attributed to the cost of using a link between two graph vertices.
- **Criterion –** It represents the choice of a link, as a function of the link weight, which insures the optimality of the graph algorithm

The *kinetic* graph theory basically uses the same definitions, but adapted to moving structures:

- **Time Varying Link Weight –** It is a continuous and integrable function related the evolution of the link weight with time. It needs to be continuous in order to insure a value for the link weight at each time instant, and also integrable as two time varying link weights are compared by their primitive integrated over the simulation time.
- **Transition –** It is the precise time at which one time varying link weight becomes better than another one.
- **Activation –** It is a time interval, between two successive transitions, during which a link is active and valid.
- **Kinetic Criterion –** It represents the choice of a *set* of links as a function of time varying link weights and activations, which insures the optimality of the kinetic graph algorithm.

Based on the previous definitions, we now describe the four steps of the Kinetic Graph Framework.

ACCEPTED MANUSCRIPT

3.2 Trajectory Knowledge

In order to model trajectories in Kinetic Graphs, we need to define the motion hypothesis in order to reduce the complexity of the kinematic model. For example, if we can assume a fixed velocity or a fixed acceleration between two trajectory changes, we may either use a first order or a second order kinematic model. The worst case scenario is if we cannot assume any kinematic hypothesis and thus must use a sophisticated stochastic prediction model. In this paper, we chose to assume a fixed velocity between two successive trajectories, and therefore used a first order prediction model possibly improved by a stochastic validity function.

According to the analysis on the prediction errors, it is obvious that a first order prediction model produces a significant adequacy error with respect to vehicular mobility, but by wisely choosing the kinetic nodal degree, a prediction criterion less sensitive to adequacy errors than the distance, we will be able to reduce the effect of the adequacy and predictability errors. Indeed, it is not because a node slightly mis-modeled its neighbors' trajectory, meaning that the neighbor is not exactly where the node thinks it should be (adequacy), or the neighbor changed its trajectory before the node thought it would (predictability), that the node needs to update its nodal degree. Finally, the realism depend on the mobility model employed for the simulations. As we chose to use VanetMobiSim, a realistic and validated vehicular mobility model, the realism error will be limited. We yet acknowledge that using the nodal degree in order to reduce the prediction error is a trick that cannot be applied to all protocols. We let the definition and use of more sophisticated stochastic kinematic models to future work.

We base our trajectory computation on Location Information, which may be provided by the Global Positioning System (GPS) or other solutions exposed in [5] or [6] and exchanged by means of beacon messages. Velocity may be derived through successive location samples at close time instants. We also assume a global time synchronization between nodes in the network which could also be obtained by the GPS system. Accordingly, we define x, y, dx, dy as the four parameters describing a node's position and instant velocity [1], thereafter called *mobility*.

Over a relatively short period of time [2], we assume that each such node, say i, follows a linear trajectory. Its position as a function of time is then described by

$$\mathbf{Pos}_i(t) = \begin{bmatrix} x_i + dx_i \cdot t \\ y_i + dy_i \cdot t \end{bmatrix}, \tag{1}$$

[1] Unless otherwise specified, we are considered moving in a two-dimensional plane.
[2] The time required to transmit a data packet is orders of magnitude shorter than the time the node is moving along a fixed trajectory.

where $Pos_i(t)$ represents the position of node i at time t, the vector $[x_i,\ y_i]^T$ denotes the initial position of node i, and vector $[dx_i,\ dy_i]^T$ its initial instantaneous velocity. Let us consider node j as a neighbor of i. In order to let node i compute node j's trajectory, let us define the squared distance between nodes i and j as

$$\begin{aligned} D_{ij}^2(t) &= D_{ji}^2(t) = \|\mathbf{Pos}_j(t) - \mathbf{Pos}_i(t)\|^2 \\ &= \left(\begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} + \begin{bmatrix} dx_j - dx_i \\ dy_j - dy_i \end{bmatrix} \cdot t \right)^2 \\ &= a_{ij}t^2 + b_{ij}t + c_{ij}, \end{aligned} \tag{2}$$

where $a_{ij} \geq 0$, $c_{ij} \geq 0$. Consequently, a_{ij}, b_{ij}, c_{ij} are defined as the three parameters describing nodes i and j mutual trajectories. And $D_{ij}^2(t) = a_{ij}t^2 + b_{ij}t + c_{ij}$, representing j's relative distance to node i, is denoted as j's linear relative trajectory to i. Consequently, thanks to (1), a node is able to compute the future position of its neighbors, and by using (2), it is able to extract any neighboring node's future relative distance.

Finally, considering r as nodes maximum transmission range, according to the Unit Disk Graph (UDG)[3], as long as $D_{ij}^2(t) \leq r^2$, nodes i and j are neighbors. Therefore, solving

$$\begin{aligned} D_{ij}^2(t) - r^2 &= 0 \\ a_{ij}t^2 + b_{ij}t + c_{ij} - r^2 &= 0, \end{aligned} \tag{3}$$

gives t_{ij}^{from} and t_{ij}^{to} as the time intervals during which nodes i and j remain neighbors.

3.3 *Neighborhood Discovery*

Basically, the *Kinetic Graph* neighborhood discovery procedure makes a node detect changes in its neighborhood without exchanges of periodical beacon messages. During this phase, each node broadcasts a single[4] $Hello$ message indicating its

[3] A Unit Disk Graph is a graph in which every two nodes are connected with an edge if and only if they are at a distance at most one. Up to normalization, a UDG corresponds to a graph where every two nodes are connected if and only if they are at a distance at most the homogeneous transmission range.

[4] In order to take into account possible collision and packet losses, a $Hello$ message is sent a configurable number of times. Unless otherwise specified, we send each $Hello$ message 3 times.

ACCEPTED MANUSCRIPT

presence in the neighborhood, and transmitting its mobility parameters. Such message is emitted using maximum transmission power in order to reach the maximum number of neighbors, and is never forwarded. Thanks to mobility predictions, upon completion of this discovery procedure, nodes in the network have an accurate knowledge of their neighborhood, and as long as their neighbors keep on moving along their initial linear trajectories, there will be no need to refresh it by sending new $Hello$ messages. If such prediction becomes invalid due to an unpredicted event (i.e. trajectory changes or disconnections), the respective node spontaneously advertises its new parameters, refreshing the predictions in a event-driven way.

In the rest of this section, we will list the content and format of geo-localization information, and then discuss the cost of transmitting such data.

3.3.1 Geo-localization Data Format

In basic simulation environments such as ns-2, Qualnet or Opnet, geo-localization data is usually based on Cartesian coordinates and the simulator's clock for time references. However, in real vehicular deployment, it is envisioned to directly use the coordinates provided by a GPS-like system (and A-GPS for indoor location), whose benefits are twofold. First, it provides a standard reference coordinates, and second, it ensures a global synchronization based on the atomic GPS clock.

The first data that the Kinetic Graph needs is a sampled position of a node. It may either be represented by Cartesian coordinates X and Y or GPS *longitude* and *latitude*, encoded in 4 bytes each. The speed vector is also crucial to a correct prediction and also needs to be included in a geo-localization message. It may either be represented by a normalized Cartesian projection of the speed vector, ie dx and dy, or by the GPS *azimuth* and *velocity*. The transmission of the speed therefore requires two coordinates encoded in 4 bytes each. Finally, time is also required in order to set the correct time scale for the prediction. A time stamp may either be sampled from the simulator's clock or by the GPS atomic clock before being transmitted. In both cases, time is encoded in 8 bytes. In total, the transmission of geo-localization information requires a transmission overhead of 24 bytes per message.

We illustrate in Figure 3 an layout example of a geo-localization message that is exchanged between two nodes implementing the Kinetic Graph Framework.

3.3.2 Discussion

Transmitting geo-localization data is a tradeoff between the potential benefits obtained by network protocols and the cost of their transmission. Indeed, it is expected that network protocols will need the geo-localization data of the sender and also of the sender immediate neighbors. Accordingly, as the network becomes dense, the

ACCEPTED MANUSCRIPT

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     HELLO     | Resv|0|0|1|0|0|   Length      |    Address    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Address                  | Resv|0|1|1|1|1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Longitude          |            Latitude           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Elevation          |    Velocity   |    Azimuth    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Stability   |              Time             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3. Hello Packet Containing Geo-localization Information

overhead induced by the transmission of these geo-localization data increases significantly. Figure 4 illustrates the cost of the transmission of geo-localization data as a function of the node degree. We can see that transmitting geo-localization without compression becomes a serious limiting factor for efficient network usage, as each packet could reach more than $1kbytes$ for dense networks. When using the compression proposed in [7], we can significantly reduce this drawback, which in turn could help improve mobility protocols in general, and Kinetic Graphs in particular.

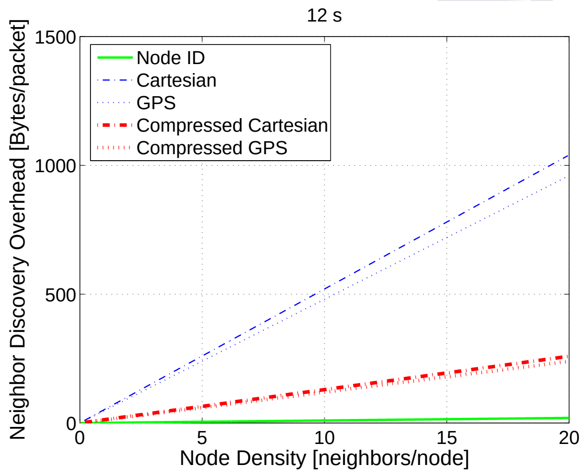


Figure 4. Illustration of the per packet overhead for geo-localization data transmission

3.4 *Time Varying Link Weights*

In this section, we describe two popular link weights used in graph theory and which could be applied to kinetic graphs. Based on those weights, a graph can be build and dynamically updated. Most of the graph algorithms could be adapted to use those criteria, however, as mentioned in the introduction part of this paper, it is important that graph protocols be distributed and local. Accordingly, we suggest as potential targets localized graph constructions described in [8].

ACCEPTED MANUSCRIPT

3.4.1 Kinetic Distance-based Weight

The *power cost* function, required to transmit between nodes i and j at time t, is defined as $P_{ij}(t) = C \cdot D_{ij}^{\alpha}(t) + \gamma$, where $\alpha \geq 2$ and for some constants γ. As we assume free space propagation and homogeneous antennas characteristics, we set $\alpha = 2$ and $C = 1$. The constant γ represents a constant charge for each transmission, including the energy needed for signal processing, internal computation, and overhead due to MAC control messages. However, since we assume perfect channel, and that the election is distributed and does not put any extra burden on any particular node, γ is common to all nodes and is not of great significance when comparing power costs. Therefore, without loss of generality, we assume $\gamma = 0$ [5] and define

$$P_{ij}(t) = D_{ij}^{2}(t) = a_{ij}t^2 + b_{ij}t + c_{ij} \tag{4}$$

as the power cost function for the weight of the *Kinetic Graphs*. By choosing the distance between nodes as the link cost, one obtains minimum power routes that help preserve battery life (see Figure 5).

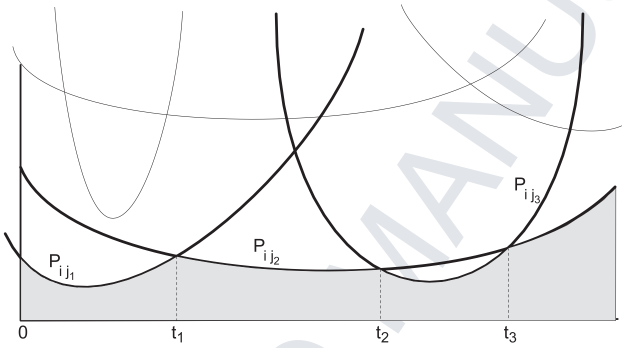


Figure 5. The power function, where each parabola represents the energy needed to reach each neighbor of node i as a function of time

We then define

$$p_i(t) = e^{-\beta_i(t-t_i)} \tag{5}$$

as the probability that a node i is continuing on its present trajectory, where the Poisson parameter $\frac{1}{\beta_i}$ indicates the average time the node follows a trajectory, and t_i the time its current trajectory has begun (see Figure 6).

Assuming independent node trajectories,

$$p_{ij}(t) = p_i(t) \cdot p_j(t) = e^{-(\beta_i+\beta_j)(t-\frac{t_i\beta_i+t_j\beta_j}{\beta_i+\beta_j})} = e^{-\beta_{ij}(t-t_{ij})} \tag{6}$$

describes the probability that nodes i and j are continuing on their respective courses at time t, which will be considered as the *stability* [6] of link $\overline{ij}$. The modi-

[5] Therefore, Power and Distance will later be interchangeably used.

[6] The probability that the mutual trajectory between two nodes remains identical after both nodes have changed course at the same time is negligible

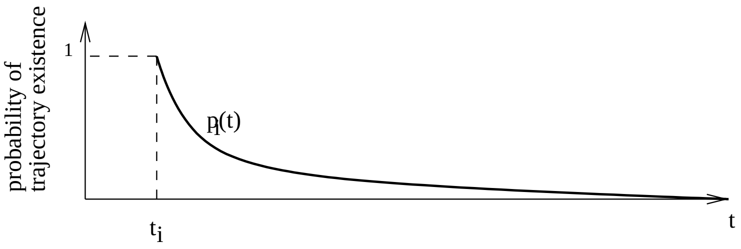


Figure 6. The stability function, where the probability for a node i to behave as predicted decreases exponentially

fied power cost below probabilistically weights the power cost $P_{ij}(t)$ to reflect the link's *stability*.

Finally, since we aim at suppressing periodic beacon messages, a node that will shortly leave the neighborhood must be automatically removed from the neighboring table. We use t_{ij}^{to} as a *timeout* counter. Upon expiration, it will remove the corresponding neighbor from the table. The link weight computed so far is able to dynamically represent the energy cost between two mobile nodes. However, it does not represent the actual capability to reach the neighbor, more specifically if two nodes stop being within mutual transmission range. For that matter, we must add a function which invalidates a link weight as soon as two neighbors stop being neighbors in the Unit Disk Graph sense. Accordingly, to represent the node's finite range, we use an inverse sigmoid function

$$Sigm_i(t) = \frac{1}{1 + e^{a\cdot(t - t_i^{to})}} \tag{7}$$

whose value is equal to 1 as long as $t < t_i^{to}$ and thereafter drops to 0, where t_{ij}^{to} is computed as described in Section 3.2.

We finally define

$$W_{ij}(t) = -\frac{p_{ij}(t)}{P_{ij}(t)} \cdot Sigm_{ij}(t) = -\frac{e^{-(\beta_{ij})(t-t_{ij})}}{a_{ij}t^2 + b_{ij}t + c_{ij}} \cdot \frac{1}{1 + e^{a\cdot(t - t_{ij}^{to})}} \tag{8}$$

$$W_{ij}(t) = -\frac{e^{-(\beta_i+\beta_j)(t - \frac{t_i\beta_i + t_j\beta_j}{\beta_i+\beta_j})}}{a_{ij}t^2 + b_{ij}t + c_{ij}} \cdot \frac{1}{1 + e^{a\cdot(t - t_{ij}^{to})}} \tag{9}$$

as the composite link weight between two neighbors (see Figure 7). A low modified power cost favors a low power cost with high stability. We have then six parameters a_{ij}, b_{ij}, c_{ij}, β_{ij}, t_{ij}, and t_{ij}^{to} describing $W_{ij}(t)$ as the time varying weight of a link between two nodes in a *Kinetic Graph*.

In order to clarify our approach, let's consider the situation depicted in Figure 8-C. Node i tries to find the best next hop node to reach a far destination node. To do so,

ACCEPTED MANUSCRIPT

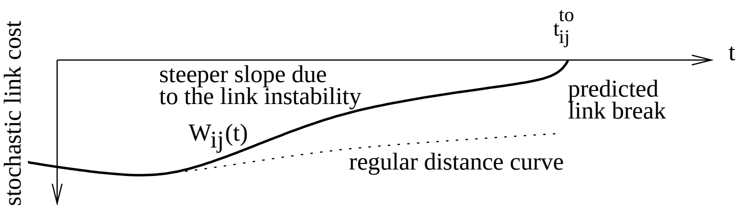


Figure 7. The composite link cost function, where we can see the cost increase due to the link's instability.

it will consider the distance separating it from its neighbors, and the stability of the respective links, in other words, the expected length of its neighbors' trajectories. Figure 8-A reflects the probabilities nodes j_1 and j_2 are not to have changed their trajectories. t_{j_1} and t_{j_2} are the time they actually began. As it can be seen, at time t_0-t_0 representing the execution time-the probability node j_1 has not to have changed its trajectory is bigger than j_2. Therefore, as depicted in Figure 8-B, even though node j_2 is closer to node i and has a similar trajectory, this link is less reliable than j_1's. However, at time t_{trans}, node j_2 has a relatively more reliable link and follows a similar trajectory that node i. Therefore, at this time, node i automatically changes its next hop neighbor, and this, without any exchange of messages.

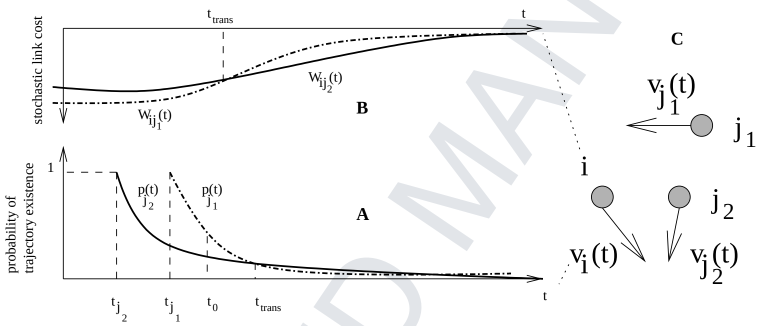


Figure 8. Topology example

3.4.2 *Kinetic Nodal Degree Weight*

In Graph theory, besides the Euclidean distance, the nodal degree is also widely used, as it provides high data spreading efficiency instead of low weight structure. While the former is popular as a criterion for routing protocol (i.e. Distance Vector), the latter is very popular for broadcast and multicast protocols, as a node with a high nodal degree has a larger diffusion potential.

Similar to the euclidean distance, the nodal degree may also be applied to Kinetic Graphs as a time varying link weight. We explain in this section, the method for modeling *Kinetic Nodal Degrees* in MANETs.

As defined in Section 3.2, we model two nodes i and j mutual trajectory as

$$D_{ij}^2(t) = a_{ij}t^2 + b_{ij}t + c_{ij} \tag{10}$$

Consequently, thanks to (10), a node is able to compute the future position of its neighbors and is able to extract any neighboring node's future relative distance.

Considering r as nodes maximum transmission range, as long as $D_{ij}^2(t) \leq r^2$, nodes i and j are neighbors. Therefore, we obtain t_{ij}^{from} and t_{ij}^{to} as the time intervals during which nodes i and j remain neighbors. Consequently, we can model nodes' kinetic degree as two successive sigmoid functions, where the first one jumps to one when a node enters another node's neighborhood, and the second one drops to zero when that node effectively leaves that neighborhood (see Figure 9).

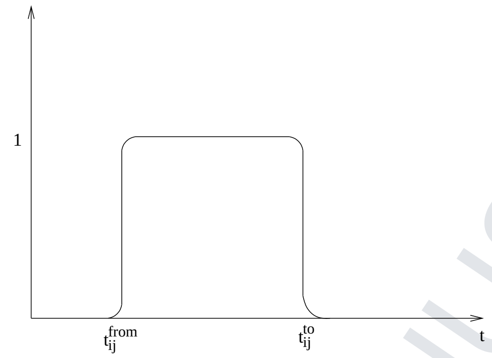


Figure 9. Double sigmoid function modeling a link lifetime between node i and node j

Considering $nbrs_i(t)$ as the total number of neighbors detected in node i's neighborhood at time t, we define

$$Deg_i(t) = \sum_{k=0}^{nbrs_i(t)} \left(\frac{1}{1+\exp(-a\cdot(t-t_k^{from}))} \cdot \frac{1}{1+\exp(a\cdot(t-t_k^{to}))} \right)$$

as node i's kinetic degree function, where t_k^{from} and t_k^{to} represent respectively the time a node k enters and leaves i's neighborhood. Thanks to (11), each node is able to predict its actual and future degree and thus is able to proactively adapt its coverage capacity. Figure 10(a) illustrates the situation for three nodes. Node k enters i's neighborhood at time $t = 4s$ and leaves it at time $t = 16s$. Meanwhile, node j leaves i's neighborhood at time $t = 20s$. Consequently, Figure 10(b) illustrates the evolution of the kinetic degree function over t.

Finally, the kinetic degree is obtained by integrating (11)

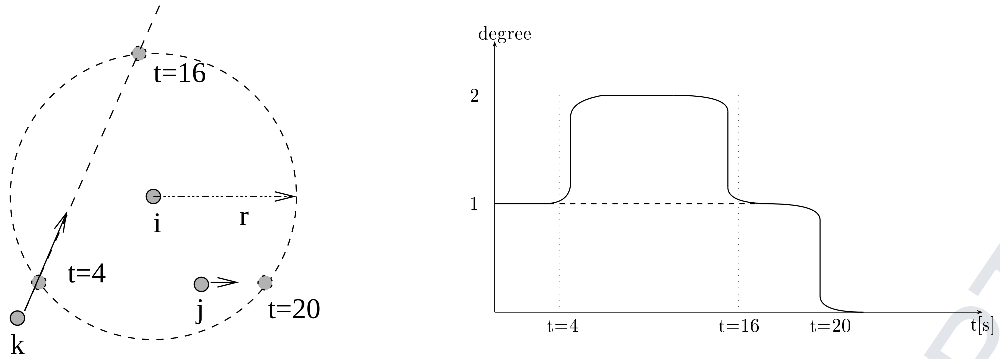
$$\widehat{Deg}_i(t) = \int_t^{\infty} \left(\sum_{k=0}^{k=nbrs_i} \left(\frac{1}{1+\exp(-a\cdot(t-t_k^{from}))} \cdot \frac{1}{1+\exp(a\cdot(t-t_k^{to}))} \right) \right) \quad (11)$$

For example, in Figure 10(b), node i kinetic degree is ≈ 32.

Similarly to the previous section 3.4.1, the kinetic nodal degree may also be stochastically weighted by the probability of the existence of the link. The last task is

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT



(a) Node i kinetic neighborhood

(b) Node i kinetic nodal degree

Figure 10. Illustration of nodes kinetic degrees

therefore to consider the uncertainty of a predicted degree by adding the stability function (6). Accordingly, we obtain a criterion reflecting nodes actual and future degree, yet biased by the uncertainty of the link between all respective neighbors.

By using substituting (6) to (11), we define

$$\widehat{Deg}_i(t) = \int_t^{\infty} \left(\sum_{k=0}^{k=nbrs_i} \left(\frac{1}{1+\exp(-a\cdot(t-t_k^{from}))} \right.\right.$$
$$\left.\left. \cdot \frac{1}{1+\exp(a\cdot(t-t_k^{to}))} \cdot \exp(-(\beta_i+\beta_k)(t-\frac{t_i\beta_i+t_k\beta_k}{\beta_i+\beta_k})) \right) \right) \quad (12)$$

1

Using the same topology as Figure 10 and applying the uncertainty of predicted degrees, we obtain a stochastically predicted nodal degree depicted in Figure 11. Initially, node i has a degree equal to 1 since node j is in its neighborhood and both initiated their trajectories at the same time. Yet, as time elapses, so does the probability both nodes have to keep their trajectories. Therefore, the stochastically predicted degree decreases. Then, at time $t = 4$, node i detects a new neighbor k and computes the time during which both nodes will be in range. However, node k initiated its trajectory before nodes i and j, consequently node k's Poisson function is smaller than node j's (see Figure 11 bottom part). Thus, during the interval node i and k are in range, the nodal degree of node i does not increase as much as it did in Figure 10. Worse, its decreasing curve is sharper than the one between nodes i and j taken alone. Similarly to Figure 10, at time $t = 16$ and $t = 20$, nodes k and j leave i's neighborhood thus making i's nodal degree decrease abruptly. The main difference here between the two figures, is that the degree is not stable during the time two nodes are in range but decreases following the probability both nodes are still following their initial trajectories.

ACCEPTED MANUSCRIPT

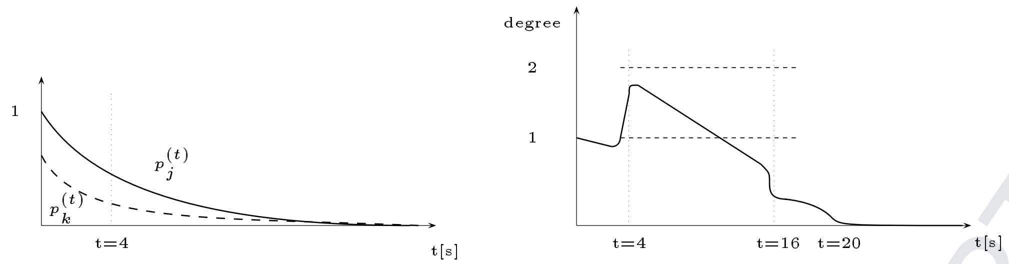


Figure 11. Stochastically Predicted Nodal Degree

3.5 *Aperiodic Neighborhood Maintenance*

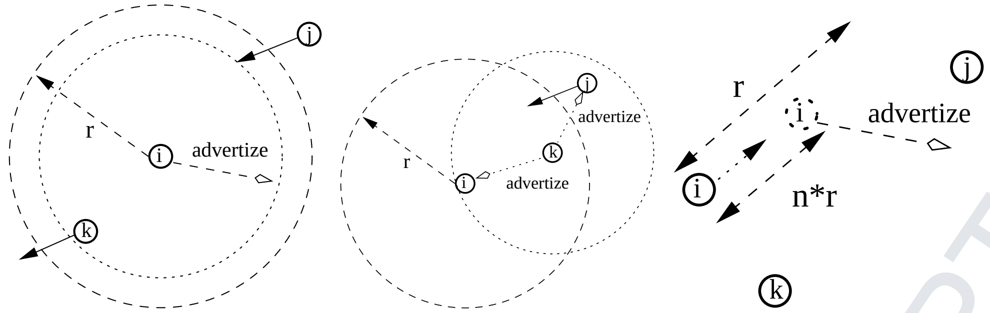
A limitation in per-event maintenance strategies is the neighborhood maintenance. While mobility prediction and the kinetic graph approach allow to discard invalid links or unreachable neighbors, it remains impossible to passively acquire new neighbors reaching some other nodes' neighborhood. The lack of an appropriate method to tackle this issue would limit Kinetic Graphs' ability to obtain up-to-date links and effective kinetic weights.

We developed several heuristics to help Kinetic Graphs detect nodes stealthily entering some other nodes transmission range in a non-periodic way.

- ***Constant Degree Detection***— Every node tries to keep a constant neighbor degree. Therefore, when a node i detects that a neighbor actually left its neighborhood, it tries to acquire new neighbors by sending a small advertising message. (see Figure 12(a));
- ***Implicit Detection***— A node j entering node i transmission range has a high probability to have a common neighbor with i. Considering the case depicted in Figure 12(b), node k is aware of both i and j's movement, thus is able to compute the moment at which either j or i enters each other's transmission range. Therefore, node k sends a notification message to both nodes. In that case, we say that node i implicitly detected node j and vice versa;
- ***Adaptive Coverage Detection***— We require each node to send an advertising message when it has moved a distance equal to a part of its transmission range. An adjusting factor which vary between 0 and 1 depends on the node's degree and its velocity (see Figure 12(c));

All three heuristics may be implemented simultaneously, further improving the capability to detect nodes stealthily entering other nodes neighborhood. The adaptive coverage contains an adjusting factor that can be tweaked. If nodes send beacons after having moved a large part of their transmission range, we reduce the beaconing overhead but also reduce the capability to detect new neighbors, whereas if they send a beacon after having moved a shorter distance, we improve the capacity to discover new neighbors at the cost of an increased beaconing overhead. In this

ACCEPTED MANUSCRIPT



(a) Constant Degree Detection (b) Implicit Detection (c) Adaptive Coverage Detection

Figure 12. Three heuristics to detect incoming neighbors in a per-event basis

paper, we chose an adjusting factor of $n = \frac{2}{3}$.

A second approach is identical to the information exchange period proposed in [9]. The idea is to determine the refreshing rate by a probabilistic model with the following assumptions:

- All nodes are randomly distributed within a disk of area $S0$ and the total number of nodes in G,N, is known.
- For a short time interval of length t, each node moves independently toward a random direction in $(0, 2\pi)$ with a constant speed v that is uniformly distributed in $[0, vmax]$.
- The maximum transmission range of a node is $d = dmax$.

Under these assumptions, Li [9] calculated the probabilities that a new neighbor moves into the transmission range of node u within a time interval of t. We ignore the case of existing neighbors moving out of the transmission range of node u since we already know this intervals.

The probability, p_{join}, that node w moves into transmission range of node u within time t is

$$\begin{cases} p_{join} = \int_d^{d+r} \frac{2xS_1}{S_0 r^2} & for\ 0 < r < 2d \\ p_{join} = \frac{\pi d^2(r-2d)}{S_0 r} \int_{d-r}^{d+r} \frac{2xS_1}{S_0 r^2} & for\ r \geq 2d \end{cases}$$

Then, given that node u has n neighbors and the total number of nodes is N. the probability that no new neighbor enters the visible neighborhood of node u is

$$p_1 = (1 - p_{join})^{N-n-1}$$

Therefore, the probability that the visible neighborhood of node u changes is

$$p_{change} = 1 - p_1$$

ACCEPTED MANUSCRIPT

Given a predetermined probability threshold p_{th}, we can determine the neighborhood update interval t such that $p_{change} < p_{th}$.

4 Application of Kinetic Graphs to Broadcasting in VANETs

Broadcasting in VANETs and ITS is a major application aimed at improving driving safety or comfort. However, insuring that broadcast packets are correctly delivered to all cars without wasting network resource is a key research objective. In this section, we propose to improve broadcasting by employing the kinetic graph approach with link weights represented by kinetic nodal degrees. We introduce the *Kinetic Multipoint Relaying (KMPR)* protocol which heuristic selects kinetic relays based on nodes actual and future predicted nodal degrees. Based on this, the topology maintenance may be limited to the instants when a change in the neighborhood actually occurs. Our objective is to show that this approach is able to adapt a broadcasting structure faster to dynamic topologies despite the dynamism of vehicular mobility, and this as a much lower maintenance cost.

We first provides a short description of the original MPR protocol and then describe the heuristics for the construction of the kinetic backbone using the Kinetic Nodal Degree as time-varying link weight. Specifications for the trajectory definition, neighborhood discovery, link weight, and aperiodic maintenance are similar to those described for Kinetic Graphs in Section 3.

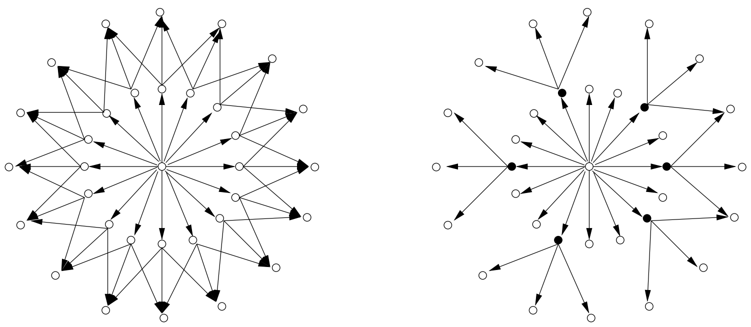
4.1 Multipoint Relays (MPR)

In order to reduce the effect of broadcasting messages to all nodes in the network, a subset of nodes, called *Multipoint Relays (MPR)*, is selected to be part of a relaying backbone. In order to build this structure, each node gathers 2-hops neighborhood information and elects the smallest number of relays such that all 2-hops neighbors are covered by at least one relay. Nodes notifies the respective relays of their decision such that each relay maintain a list of nodes, called *Multipoint Relaying Selectors (MPR Selector)*, which has elected it as MPR. Finally, the relaying decision is made on the basis of last-hop address according to the following rule:

Definition 8 (MPR flooding) *A node retransmits a packet only once after having received the packet the first time from a MPR selector.*

Figure 13 shows a node with its set of 1-hop and 2-hops neighbors. Figure 13(a) depicts the initial full topology, while Figure 13(b) illustrates the MPR topology, where solid circles are MPRs to the central nodes. Accordingly, the central node is part of the MPR Selector list of each solid circles nodes.

ACCEPTED MANUSCRIPT



(a) All neighbors retransmit broadcast (b) Only MPR nodes relay broadcast

Figure 13. Illustration of flooding reduction using MPR

4.2 Kinetic Multipoint Relays (KMPR)

To select the kinetic multipoint relays for node i, let us call the set of 1-hop neighbors of node i as $N(i)$, and the set of its 2-hops neighbors as $N^2(i)$. We first start by giving some definitions.

Definition 1 (Covering Interval) *The covering interval is a time interval during which a node in $N^2(i)$ is covered by a node in $N(i)$. Each node in $N^2(i)$ has a covering interval per node i, which is initially equal to the connection interval between its covering node in $N(i)$ and node i. Then, each time a node in $N^2(i)$ is covered by a node in $N(i)$ during a given time interval, this covering interval is properly reduced. When the covering interval is reduced to $\emptyset$, we say that the node is fully covered.*

Definition 2 (Logical Kinetic Degree) *The logical kinetic degree is the nodal degree obtained by (11) but considering covering intervals instead of connection intervals. In that case, t_k^{from} and t_k^{to} will then represent the time interval during which a node $k \in N^2(i)$ starts and stops being covered by some node in $N(i)$.*

Algorithm 1 Kinetic Multipoint Relaying (KMPR)

Require: Begin with an empty KMPR set.

1: Compute the logical kinetic degree of each node in $N(i)$.

2: Add in the KMPR set the node in $N(i)$ that has the maximum logical kinetic degree. Compute the activation of the KMPR node as the maximum covering interval this node can provide. Update all other covering intervals of nodes in $N^2(i)$ considering the activation of the elected KMPR, then recompute all logical kinetic degrees. Finally, repeat this step until all nodes in $N^2(i)$ are fully covered.

The basic difference between MPR and KMPR is that unlike MPR, KMPR does not work on time instants but on time intervals. Therefore, a node is not periodically elected, but is instead designated KMPR for a time interval. During this interval,

ACCEPTED MANUSCRIPT

we say that the KMPR node is active and the time interval is called its activation.

The KMPR protocol elects a node as KMPR a node in $N(i)$ with the largest logical kinetic degree. The activation of this KMPR node is the largest covering interval of its nodes in $N^2(i)$.

Then, each node having elected a node KMPR for some activations is then a KMPR Selector during the same activation. Finally, *KMPR flooding* is defines as follows:

Definition 3 (KMPR flooding) *A node retransmits a packet only once after having received the packet the first time from an active KMPR selector.*

4.3 Performance Evaluation

We implemented the KMPR protocol under ns-2.29 and used the NRL-OLSR [10] implementation for comparison with KMPR. We measured several significant metrics for VANETs: The effectiveness of flooding reduction, the delay before the network receives a broadcast packet, the number of duplicate packets and finally the routing overhead. We used a square simulation area of 2000×2000 with a node density of $8nbrs/node$. For realistic results, we used VanetMobiSim, where we increased the road segment length from $50m$ to $200m$ [7]. As we wanted to illustrate the effect of mobility, we did not include pause time at the end of a trip. Finally, we simulated the system at steady state for $100s$, and each point is plotted with a 95% confidence interval.

As the objective is to evaluate KMPR with respect to an increasing dynamism of the network, the straightforward choice is to increase the speed. Although this remark is straightforward for random mobility, realistic vehicular mobility patterns are an issue. Indeed, the real velocity effectively reached by a car cannot be configured apriori, as it is subject to external influences as illustrated in Figure 1. So, if we cannot control the velocity, we may influence the parameters that control it. As it has been observed in [11], the length of a road segment between two successive intersection has a significant impact on the real speed reached by cars, as cars are required to reduce their speed and stop at each intersection. Accordingly, when increasing the dynamism of vehicular networks, we will actually increase the road segment length.

In this section, we also illustrate the impact of the prediction errors on the Kinetic Graphs. For the adequacy error, we compare the efficiency of the Kinetic Graphs using random and vehicular mobility, using respectively the Random Waypoint Mobility (RWM) model and VanetMobiSim [4]. As the trajectories generated by

[7] As shown in [11], increasing the length between two intersections is a more significant method in order to increase the real vehicular speed than the configured average speed.

the RWM are in total adequacy with the ones used by the framework, the adequacy error is canceled in the case of random mobility. According to the framework, we also nullified the predictability error by updating the structure at each predictability interval, yet as the cost of a maintenance strictly tided to the predictability interval. Therefore, an increasing topology dynamism from an increased speed or shortened road segment lengths will reduce the predictability interval and thus increase the maintenance cost of the kinetic structure. Finally, as already mentioned before, by using a realistic vehicular mobility model instead of only random mobility, we will also be able to significantly reduce the realism errors in our evaluations.

A general preliminary remark is that the MPR protocol, which has been obtained from graph theory, is able to build a connected dominating set in *perfect conditions*. In other words, on a static network and when the wireless channel is perfect, i.e. each broadcast packet is corrected received by all neighbors within the intended communication range, the subset of relays created by MPR form the optimal relay set required by a broadcast packet to be correctly delivered to the entire network. Unfortunately, the *perfect conditions* assumption is wrong in practice, and mobility added to a challenging wireless channel make the MPR set significantly deviate from its initial optimality. Either more relays are designated that are really necessary, or conversely, relays are missing and thus the relay set becomes disconnected. In any case, there is not much the original MPR algorithm is able to do in order to reduce this issue, as it faces the same limitations as all protocol developed for mobile wireless networks. As previously discussed, the objective of the Kinetic Graph framework is to suppress the periodic maintenance that in turn improves the broadcast channel's reliability. In the rest of this section, we illustrate how the relay set based on actual and future topologies and not on past ones, and also by improving the broadcast channel efficiency, the KMPR protocol is able to reduce the deviation of the MPR protocol from the optimal MPR set in perfect conditions.

Figure 14(a) and in Figure 14(b) first illustrate forwarding packet ratio representing the size of the MPR set. According to the original MPR protocol, the optimal number of MPR relays, MPR^{opt} is proportional to $\sqrt[3]{n}$, where n is the number of nodes in the network, or more generally, the density of neighbors per node. As the density used for these simulations is 8 neighbors/node, the optimal number of MPR relays is $MPR^{opt} = ForwardPacketRatio^{opt} = 2$. As it may be observed in the two figures, both the MPR and KMPR sets are larger than the optimal value. However, KMPR is able to significantly reduce the gap between the MPR set and the optimal MPR set. This reduction is particularly exacerbated in the case of random mobility, where the adequacy is perfect. When considering adequacy errors with vehicular mobility, KMPR also deviates from the optimal set, but still remains better than MPR. The impact of the predictability on the KMPR set is limited thanks to the nodal-degree criterion in both figures. Accordingly, we can see that the Kinetic Graph framework is well adapted to highly mobile topologies when building and maintaining a structure such as a relay set.

ACCEPTED MANUSCRIPT

On Figure 15(a) and in Figure 15(b), we further analyze the broadcast efficiency of MPR and KMPR with the duplicate reception ratio. The ratio has been computed as the number of packets duplicated per packet sent. Clearly, a duplicate packet is useless in term of broadcast efficiency when the broadcast network is connected. Accordingly, the smaller the duplicate ratio is, the better is the broadcast efficiency. Similarly to the forward packet ratio, KMPR is able to significantly reduce the number of duplicate packets that are carried in the wireless network compared to MPR. As the decision to forward a packet, and possibility generate a duplicate packet, also depends on the relay set, KMPR manages to keep a more accurate relay set with respect to the instantaneous topology. By comparing the duplicate ratio between random and vehicular motions, i.e with perfect and reduced adequacy, we can clearly see the effect of the adequacy error, as the duplicate ratio is increased by a factor of 40%. As it may also be observed, the duplicate ratio is affected by a reduced predictability, a feature that is specific to the MPR connected dominating set. Indeed, in standard connected dominating sets (CDS), a relay node forwards any packet in its queue that it sees for the first time. Therefore, as the KMPR relay set is not influenced by the predictability interval, neither should the duplicate ratio. Yet, a (K)MPR-CDS node forwards a packet based on the last hop information (the node that elected it as relay). Therefore, a difference between the topology based on which a relay has been elected and the current topology has an impact. And due to the reduced predictability, thus an increased maintenance rate, the new (K)MPR set cannot be updated on time for the a perfect adequacy case, or the set is updated on wrong topology information for a reduced adequacy. In other words, there is a deprecancy between the nodes that elected a (K)MPR node and the (K)MPR node itself. Luckily, KMPR reduces this deprecancy compared to MPR, as it manages to reach respectively a 40% and 10% reduction in the duplicate ratio with perfect adequacy and reduced adequacy. Indeed, KMPR bases the election on the current state information and not on the past, and also significantly reduces the maintenance cycles as we will show in Figure 17(a) and in Figure 17(b). One way for the Kinetic Graph framework to further reduce the influence of the predictability is to increase the predictability interval, in other words, to develop a more accurate prediction model than the linear first order model, even in the case of random motion. That is part of future works.

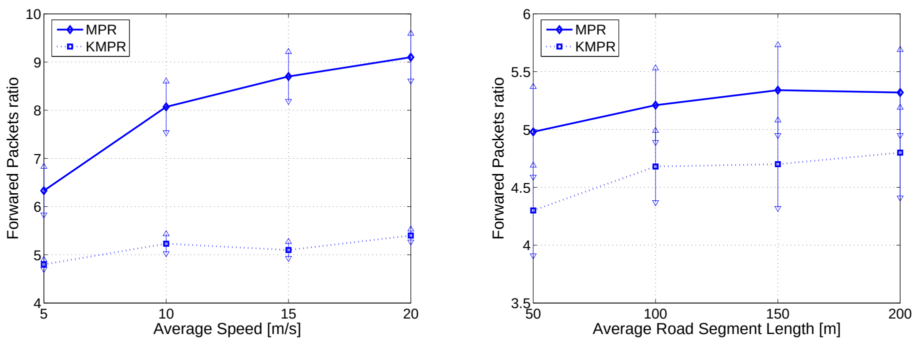
Second, we illustrate in Figure 16(a) and in Figure 16(b) the end-to-end delivery delay, which represents the time required to successfully deliver a broadcast packet to the network. We artificially only consider the delivery delay of packets that may be correctly delivered on the first broadcast attempt. Indeed, we are interested in measuring the efficiency of the *Connected Dominating Set* formed by the MPR, resp. KMPR nodes, and thus neglect dropped packets and successive retransmission attempts. Nodes that do not receive the first copy of a broadcast packet form an *Unconnected Dominating Set*. We can clearly see on both figures that the delay is significantly reduced when using kinetic graphs. Two reasons are behind this feature. As KMPR builds its topology on actual and future configuration, the structure is always adapted to the correct topology. Second, as periodic maintenance

ACCEPTED MANUSCRIPT

messages are not sent, the channel is more available for data traffic. And as we will show later, the ratio of nodes to which a broadcast packet is delivered is also significantly bigger than with MPR.

We may then see on Figure 17(a) and in Figure 17(b) KMPR's maintenance overhead in bytes compared to MPR. As a reminder, the maintenance overhead is measured by computing the ratio of the number of bytes required for the maintenance with the total number of bytes (traffic and maintenance) transferred on the wireless network. We can see that by using the Kinetic Graph framework, KMPR is able to significantly reduce the routing overhead ratio. Yet, by increasing the speed or the road segment length, and thus reducing the predictability interval, KMPR's routing overhead ratio grows and even overpasses MPR's. Indeed, as the Kinetic Graph framework updates the structure at each predictability interval, the number of messages is worsen. Moreover, each kinetic message being bigger due to the transmission of position information and other kinematics data, the overhead in bytes is furthermore degraded compared to the packet overhead. There are therefore configurations, where the non-periodic approach might look better in term of maintenance overhead, but as we previously showed, this does not apply to the other broadcast efficiency metrics, such as a kinetic protocol benefits from an enhanced topology knowledge compared to the non-kinetic approach.

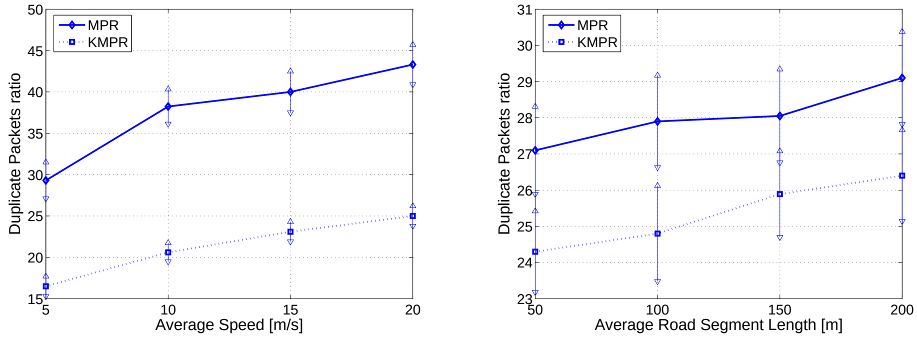
Finally, Figure 18(b) illustrates the *Unconnected Dominating Set Ratio*. All results obtained so far have been averaged over tests when *all* nodes could obtain a copy of the broadcast packet (we discarded tests where at least one node could not get the broadcast packet). Due to the particular spatial distribution of cars creating clusters at intersections and a sparse connectivity in between for instance, frequent disconnections occur for MPR. This figure therefore illustrates the ratio of runs which showed a disconnected graph with respect of the total number of runs. It is straightforward from this figure to see that, as KMPR is less sensitive to mobility and adapts to dynamic topologies much faster, these temporal or spatial disconnections are significantly reduced, further improving the reliability of KMPR for broadcasting in VANETs.



(a) Perfect Adequacy (random) (b) Reduced Adequacy (vehicular)

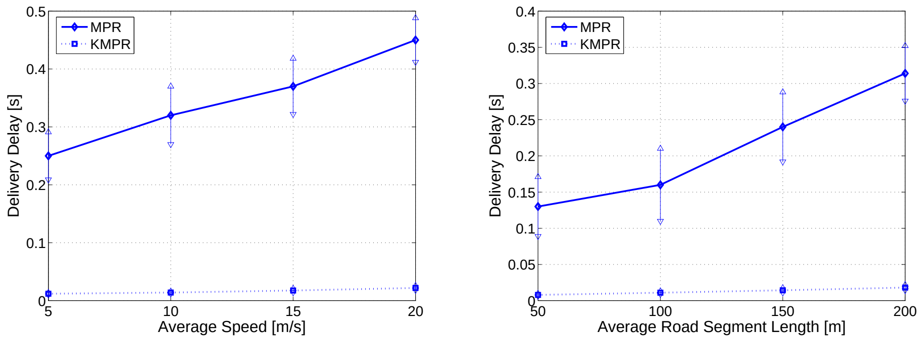
Figure 14. Illustration of Forwarding Ratio vs. Predictability

ACCEPTED MANUSCRIPT



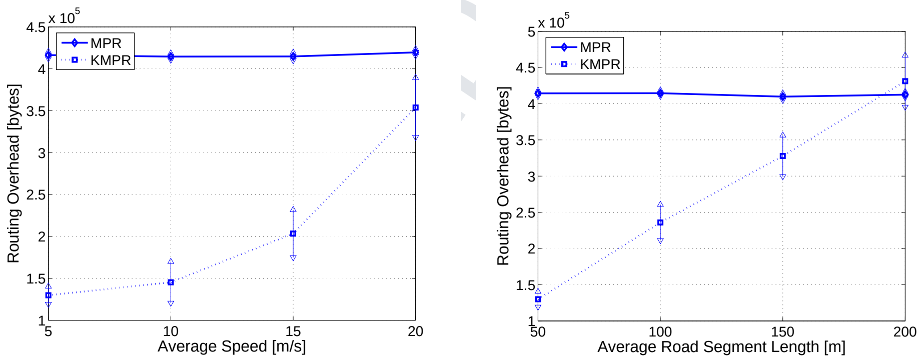
(a) Perfect Adequacy (random) (b) Reduced Adequacy (vehicular)

Figure 15. Illustration of Duplicate Packet Ratio vs. Predictability



(a) Perfect Adequacy (random) (b) Reduced Adequacy (vehicular)

Figure 16. Illustration of End-to-End Delay vs. Predictability

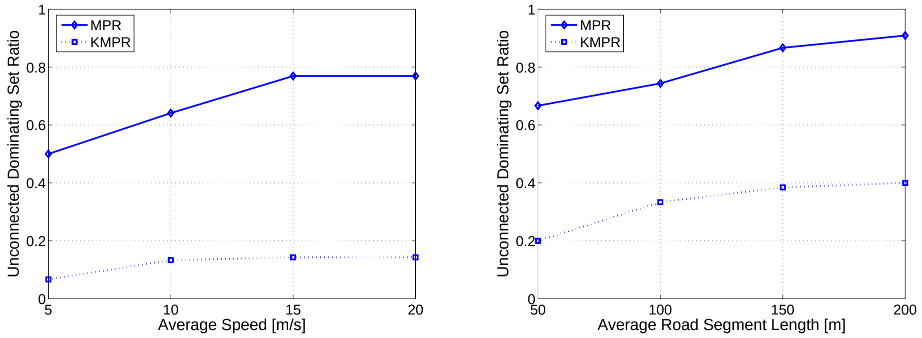


(a) Perfect Adequacy (random) (b) Reduced Adequacy (vehicular)

Figure 17. Illustration of Maintenance Overhead vs. Predictability

In this section, we illustrated how MPR could be successfully improved by the use of Kinetic Graphs and use a Kinetic Mobility Management. We evaluated the influence of prediction errors, such as realism, adequacy and the predictability error, on the performance of the Kinetic Graphs. We compared KMPR's behavior when considering the Random mobility which has a low adequacy error but a high realism error, and VanetMobiSim that has a high adequacy error but a low realism error. We also increased the predictability error by increasing the mobility of the ad hoc

ACCEPTED MANUSCRIPT



(a) Perfect Adequacy (random) (b) Reduced Adequacy (vehicular)

Figure 18. Illustration of Unconnected Dominating Set Ratio vs. Predictability

network and emphasized the relative stability of KMPR. We finally showed how even by suffering from a high adequacy error, the use of the Kinetic Degree instead of a Kinetic Distance was able to limit the scope of the prediction error. KMPR and the Kinetic Graph Framework therefore showed to be particularly adapted to highly mobile networks.

We chose not to test KMPR with respect to the network density, as the problematic of this work is mobility and not density or scale. As the number of MPR nodes for a static configuration behaves as $O(\sqrt[3]{density})$, and as MPR is a degenerated case of Kinetic MPR, we expect the scale of KMPR to be similar to MPR.

5 Related Work

Mobility management has been very early seen as a critical requirement by any network protocol when confronted to mobile terminals. In cellular networks, mobility management aims at tracking users in order for calls or other cellular services to be corrected delivered to them. It has been later extended to track users on-calls in order to prepare a hand-off.

The very first and basic mobility management technique was the periodic and proactive mobility management. As no apriory knowledge of the evolution of the topology could be assumed, a maintenance process was periodically triggered. Very early, the cellular network community understood that this process consumed a large amount of an already scarce network resource and should therefore be improved. For example, the cellular system created the hierarchical mobility management to track off-call users. When a cell-phone does not need to be precisely tracked in a base-station micro-cell (i.e. during a call), then the maintenance is limited to a macro-cell called location area.

Yet, each micro-cell may contain a large number of cellphones that need to be

ACCEPTED MANUSCRIPT

tracked as precisely as possible in order to know when they will change cell and avoid to interrupt the call. Cellular networks therefore provided the majority of related work in mobility management for wireless communications as it responded to an industrial need. Since the knowledge of the trajectory followed by a cell phone would let the network anticipate a hand-off and reduce the maintenance overhead, the most advanced solution for intra-cell mobility management were also based on mobility prediction techniques. As a matter of fact, the majority of works on the application of mobility prediction techniques has been done on cellular networks and Wireless ATMs. Various solutions have been developed, varying from user movement history, Kalman or Particle filters, or neural networks. A review of the state-of-the-art of mobility prediction techniques applied to cellular networks may be found in [12].

At the early stage of mobile ad hoc networks, mobility has been widely ignored, as the major concern of the community was scalability with respect to density. Yet, in recent years and following a similar path as that of cellular network, techniques have been developed to replace the proactive with an adaptive mobility management. Successful applications have been proposed for topology management [13,14], link availability [15,16,17,18,19], route availability [20,21,22], location services [23,24] or geo-routing [25]. In each case, the objective was to adapt the periodic maintenance to the dynamism of the mobile network, or to choose a link or a path depending of its maximum life duration.

Despite the fruitful results in adaptive mobility management, the next step to a fully reactive mobility management was never walked, or more precisely, was but has widely gone unnoticed by the MANET community. Indeed, it had been first described in the field of data structures when confronted to mobile objects. 10 years ago, the concept of *Kinetic Data Structures (KDS)* was introduced by *Bash et al.* [26], which is to the best of our knowledge the first description of an application of a reactive mobility management technique. This topic has then been widely studied in various areas such as mobile facility locations [27], clustering and routing [28], or shortest path [29]. A survey on KDS can be found in [30]. Unfortunately, at the time of this fruitful developments, the powerful distributed graph algorithms were not available, and all protocols were developed for centralized protocols.

By taking a step back and looking at the KDS and MANET community achievements, we see that the former created solutions optimally adapted to mobility, but which could not be applied to a fully distributed network, while the latter developed efficient distributed graph algorithms, which were not adapted to mobility. It therefore appeared to us straightforward that the two approaches were mutually profitable. This observation gave then birth to the *Kinetic Mobility Management* and the Kinetic Graph framework that we presented in this paper. Unlike any other mobility management developed in cellular networks or in MANETs, *Kinetic Mobility Management* is fully reactive and has also been designed to be implemented

ACCEPTED MANUSCRIPT

by any distributed graph algorithm.

6 Conclusion

As the bad performance of standard MANET protocols in vehicular networks is mostly coming from inefficient mobility management solutions, this paper focused on improving mobility management in vehicular ad hoc networks using *Kinetic Graphs* and mobility predictions. The objective was to offer an alternative to the development of specific mobility protocols for VANET and ITS. Indeed, the use of standard MANET protocols in vehicular networks is a key issue for industrial partners and standardization bodies involved in the deployment of VANET and ITS, as it would ease the interoperability between fixed networks, MANETs, and vehicular networks.

We first described the challenges of predicting mobility in vehicular networks and then provided guidelines for adapting mobility protocols to kinetic mobility management. By following these guidelines, standard protocols for MANET may be efficiently operated for vehicular networks, as it suppresses the periodic beaconing process widely used by almost all mobility protocols, and also increases the time intervals during which the mobile topology is correctly anticipated. As an application example, we depicted the improved broadcast efficiency of the *Kinetic Multipoint Relaying (KMPR)* protocol compared to the original MPR protocol. This approach is therefore able to efficiently maintain a communication backbone for VANETs and ITS, despite the dynamism of vehicular mobility.

In this work, we chose to use have a high realism but a bad adequacy with respect to modeling and predicting vehicular mobility. Yet, we chose a criterion less sensitive to prediction errors. An interesting extension could be to obtain a good realism and a good adequacy. For this matter, more sophisticated prediction models for vehicular motions should be devised. Moreover, the approach is independent of the criteria used to build the backbone and various approaches may be combined and tested. Other mobility protocols could therefore be adapted to kinetic mobility management.

References

[1] V. I. I. (VII), `http://www.its.dot.gov/vii/`.

[2] C.-.-C. C. C. (C2C-CC), `http://www.car-to-car.org/`.

[3] W. A. for the Vehicular Environment (WAVE), `http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm`.

ACCEPTED MANUSCRIPT

[4] T. V. M. S. P. (VanetMobiSim), http://vanet.eurecom.fr.

[5] R. Fontana, S. Gunderson, Ultra-wideband precision asset location system, in: Proc. of the IEEE Conference on Ultra Wideband Systems and Technologies, 2002.

[6] C. Gentile, L. Klein-Berndt, Robust location using system dynamics and motion constraints, in: Proc. of the IEEE International Conference on Communications (ICC), 2004.

[7] J. Haerri, F. Filali, C. Bonnet, Rethinking the overhead of geo-localization information for vehicular communications, Technical Report 07-194, Institut Eurécom (2007).

[8] X. Li, I. Stojmenović, Broadcasting and topology control in wireless ad hoc networks, in: A. Boukerche, I. Chlamtac (Eds.), Handbook of Algorithms for Mobile and Wireless Networking and Computing, CRC Press, 2006, Ch. 11, pp. 239–264.

[9] N. Li, J. Hou, L. Sha, Design and analysis of an mst-based topology control algorithm, in: Proc. of the IEEE INFOCOM, 2003.

[10] N. R. L. O. (NRLOLSR), http://pf.itd.nrl.navy.mil/projects.php?name=olsr.

[11] J. Härri, F. Filali, C. Bonnet, On meaningful parameters for routing in vanets urban environments under realistic mobility patterns, in: 1st IEEE Workshop on Automotive Networking and Applications (AutoNet'06) (in conjuction with IEEE Globecom 2006), 2006.

[12] R. C. Doss, A. Jennings, N. Shenoy, A review of current work on mobility prediction in wireless networks, in: 3rd ACM Asian International Mobile Computing Conference (AMOC'04), 2004.

[13] J. Härri, C. Bonnet, A lower bound for vehicles' trajectory duration, in: Proc. of the 62nd IEEE Vehicular Technology Conference (VTC'05), 2005.

[14] J. Härri, N. Nikaein, C. Bonnet, Trajectory knowledge for improving topology control in mobile ad-hoc networks, in: Proceedings of the 1st ACM/e-NEXT International Conference on Future Networking Technologies (Co-NEXT'05), 2005.

[15] H. Menouar, M. Lenardi, F. Filali, A movement prediction-based routing protocol for vehicle-to-vehicle communications, in: Proc. of the 1st International Vehicle-to-Vehicle Communications Workshop (V2V-COM'05), 2005.

[16] N. Wang, S. Chang, A reliable on-demand routing protocol for mobile ad hoc networks with mobility prediction, Elsevier Computer Communications 29 (2005) 123–135.

[17] W. Su, S. Lee, M. Gerla, Mobility prediction in wireless networks, in: IEEE Military Communication Conference (MILCOM), Vol. 1, 2000, pp. 491–495.

[18] A. Agarwal, S. R. Das, Dead reckoning for mobile ad hoc networks, in: Proc. of the 2003 IEEE Wireless Communications and Networking Conference (WCNC'03), 2003.

[19] S. Jiang, D. He, A prediction-based link availability estimation for routing metrics in manets, IEEE/ACM Transaction on Networking 13 (6) (2005) 1302–1312.

ACCEPTED MANUSCRIPT

[20] C. Doss, R. A. Jennings, N. Shenoy, Mobility prediction based routing for minimizing control overhead in mobile ad hoc networks, in: Proc. of the International Conference on Wireless Networks (ICWN'04), 2004.

[21] M. Musolesi, S. Hailes, C. Mascoloy, Adaptive routing for intermittently connected mobile ad hoc networks, in: Proc. of the 6th IEEE IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'05), 2005.

[22] M. Fiore, J. Harri, F. Filali, C. Bonnet, Vehicular mobility simulation for vanets, in: Proc of the 40th IEEE/SCS Annual Simulation Symposium (ANSS'07), 2007.

[23] S. S. *et al.*, A comparative study of mobility prediction schemes for gls location service, in: Proc. of the IEEE Vehicular Technology Conference (VTC'04)Conference, 2004.

[24] X. Luo, T. Camp, W. Navidi, Predictive methods for location services in mobile ad hoc networks, in: Proc. of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), 2005.

[25] T. K. *et al.*, Dead-reckoning for position-based forwarding on highways, in: Proc. of the 3rd International Workshop on Intelligent Transportation (WIT 2006), 2006.

[26] J. B. *et al.*, Data structures for mobile data, Journal of Algorithms 31 (1) (1999) 1–28.

[27] S. B. *et al.*, Mobile facility location, in: Proc. of the 4th international Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 2000.

[28] J. Herschberger, Smooth kinetic maintenance of clusters, in: Proc. of the 19th Annual Symposium on Computational Geometry, 2003.

[29] C. Gentile, J. Härri, R. E. V. Dyck, Kinetic minimum-power routing and clustering in mobile ad-hoc networks, in: Proc. of the IEEE Vehicular Technology Conference (VTC'02 Fall)Conference, 2002.

[30] L. Guibas, Kinetic data structures: A state of the art report, in: Proc of the 3rd Workshop of Algorithmic Foundations of Robotics, 1998.