ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

MODELING AND PREDICTING MOBILITY IN WIRELESS AD HOC NETWORKS

Jérôme Härri

Pour Obtenir le Grade de

Docteur ès Science
de l'École Polytechnique Fédérale de Lausanne (EPFL)

Date de Soutenance: 22 Juin 2007
Composition du Jury:

Prof. Bixio Rimoldi	Président
Prof. Jean-Yves Le Boudec	Rapporteur
Prof. Mario Gerla	Rapporteur
Prof. Hannes Hartenstein	Rapporteur
Prof. Christian Bonnet	Directeur de Thèse
Prof. Fethi Filali	Co-Directeur de Thèse

Thèse réalisée au sein de l'Institut Eurécom

AUTHOR'S ADDRESS:
Jérôme Härri
Institut Eurécom
Département Communications Mobiles
2229, route des Crêtes
B.P. 193
06904 Sophia-Antipolis – France
Email: Jerome.Haerri@eurecom.fr
URL: http://www.eurecom.fr/people/haerri.en.htm

To my Parents, without whom nothing would have been possible

A mes Parents, sans lesquels rien n'aurait été possible

Acknowledgments

I would like to express my most sincere gratitude to my advisor, Professor Christian Bonnet, who gave me the opportunity to work on this motivating problem. I would also like to thank him for his constant support and enlightening advice during my Doctoral Thesis.

I am also thankful to my co-advisor Professor Fethi Filali for his constant availability and help when I was at lost.

I am grateful to the members of my Doctoral Jury, Prof. Bixio Rimoldi, Prof. Jean-Yves Le Boudec, Prof. Mario Gerla and Prof. Hannes Hartenstein, for having agreed to evaluate my work.

I would like to thank Professor Mario Gerla for having invited me to work with him during summer 2006, as well as all members of the Network Research Lab at the University of California for their friendship and the good time I had during my stay.

I would also like to thank Marco Fiore, the co-creator of VanetMobiSim, for our fruitful collaboration on vehicular mobility modeling. I wish him good luck for the end of his Ph.D. thesis.

Many thanks to my colleagues, among them Dr. Melek Önen, Moritz Steiner, Saad Kiani, Ruben De Francisco, for their comments and suggestions on this manuscript.

Also many thanks to all colleagues, staff members, present and former Ph.D. students, and friends at the Institut Eurécom for the nice time I had during the last four years.

Finally, I could not forget to thank Dr. Gentile and Dr. Moayeri from the National Institute of Standards and Technology for having shared with me the initial concept that gave birth to this work, and for having transmitted me the passion for research during my Master Thesis.

My special gratitude goes to my mother Veronique, my father Ferdinand, and my sister Sophie for their constant love, support and encouragement.

Please enjoy your reading !

Abstract

Wireless Ad Hoc Networks are a particular paradigm where wireless devices communicate in a decentralized fashion, without any centralized infrastructure or decision. In order to avoid a situation where nodes chaotically try to communicate, distributed and localized structures (graphs, trees, Ě) need to be built. Mobility brings challenging issues to the maintenance and to the optimality of such structures. In conventional approaches, structures are adapted to the current topology by each node periodically sending beacon messages, which is a significant waste of network resources. If each node can obtain some a priori knowledge of future topology configurations, it could decide to send maintenance messages only when a change in the topology effectively requires updating the structure.

In this Doctoral Thesis, we investigate this approach and define the Kinetic Graphs, a novel paradigm regrouping mobility predictions for a kinetic mobility management, and localized and distributed graph protocols to insure a high scalability. The Kinetic Graph framework is able to naturally 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. By following this framework, we show that any structure-based ad-hoc protocol may benefit from the kinetic approach.

A significant challenge of Kinetic Graphs comes from prediction errors. In order to analyze them, we illustrate the relationship between the prediction model and the mobility model. We decompose the prediction errors into three metrics: the adequacy between the prediction and the mobility models, the predicability of the mobility model, and the mobility modelŠs realism. Following the framework, we define a kinetic model for the modeling of the trajectories and then analyze the extents of the effects of each error metric and develop solutions in order to reduce them. We finally adapt the Multipoint Relaying (MPR) protocol, used by the Optimized Link State Routing protocol (OLSR), and show the significant improvements that may be obtained by using the Kinetic Graph Framework, even on the very challenging vehicular networks.

Keywords — Kinetic graphs, mobility management, mobility modeling, mobility predictions, multipoint relays (MPR), optimized link state routing (OLSR), broadcasting, routing, wireless ad hoc networks, vehicular networks.

Résumé

Les nœuds d'un réseau ad hoc sans fil communiquent a priori de manière non coordonnée et décentralisée. Afin d'éviter une situation où les nœuds essaieraient de communiquer de manière désorganisée, des structures (graphes, arbres,Ě) distribuées et localisées sont construites. Cependant, la mobilité est un défi pour la gestion optimale de telles structures. Dans les méthodes conventionnelles, les structures sont adaptées à la topologie courante par chaque nœud en émettant périodiquement un message de contrôle. Cette approche génère un gaspillage de ressource réseau. Si chaque nœud réussit à obtenir une connaissance a priori de l'évolution de la topologie, les messages de contrôle peuvent être émis uniquement lors d'une modification de la topologie nécessitant une mise à jour des structures.

Dans cette thèse de doctorat, nous avons investigué cette approche et avons développé les Graphes Cinétiques, qui sont un nouveau paradigme basé sur la prédiction de mobilité. Les Graphes Cinétiques sont capables de capturer la dynamicité des structures mobiles. Le cadre de leur développement est composé de quatre étapes (i) une représentation de la trajectoire des nœuds mobiles, (ii) un format et une structure commune de transmission de ces trajectoires, (iii) des poids de liens temporellement variables, (iv) une gestion apériodique du voisinage. En suivant ce cadre, nous montrons que tout protocole réseau ad hoc basé sur des structures peut bénéficier de l'approche cinétique.

Un défit majeur des Graphes Cinétiques provient d'erreurs de prédictions. Afin de les étudier, nous illustrons la relation entre modèle de prédiction et modèle de mobilité. Nous décomposons ces erreurs de prédictions en trois métriques : l'adéquation entre le modèle de prédiction et le modèle de mobilité, la prévisibilité du modèle de mobilité, et finalement le réalisme du modèle de mobilité. En suivant le cadre de développement des graphes cinétiques, nous définissons un modèle cinétique pour la modélisation des trajectoires, analysons l'étendue de chaque métrique d'erreur, et ensuite développons des solutions afin d'en limiter les effets. Nous adaptons finalement le protocole des Relais Multipoints (MPR), principalement utilisé par le protocole OLSR, et montrons les améliorations significatives des performances réseaux qui peuvent être atteintes en utilisant les graphes cinétiques, même dans le cadre complexe des réseaux véhiculaires.

Mots-clés — Graph Cinétique, gestion de mobilité, modélisation de mobilité, prédiction de mobilité, relais multipoint (MPR), OLSR, diffusion, routage, réseaux ad-hoc sans fil, réseaux véhiculaires.

TABLE OF CONTENTS

CHAPTER I

Introduction

TWO centuries ago, eminent scientists such as Morse, Meucci, Bell or Marconi placed the first stone to a revolutionary technology allowing to transmit information at light speed to a distant receiving end. They have been the Founding Fathers of what we nowadays commonly call telecommunication. Grouped in networks in order to improve their capacity and resilience, telecommunication systems have been a milestone in our society, and maybe one of the most significant achievement of the last two centuries. Whereas the everlasting increasing vision of our world amplified our natural need for mobility and exploration, those networks paradoxically allowed us to stay virtually close.

At the end of the eighties, this evolution reached a breaking point, where drastic changes occurred in the telecommunication area. We entered into the digital era and this period also saw the birth of personal wireless mobile telecommunication systems. As hard as finding the answer of Plutarch's famous paradox about the hen and the egg, it is unclear whether mobility created the need for wireless mobile communication systems, or if those systems, originally designed for an elite, drastically and irremediably altered the society. In any case, wireless mobile communication systems became defacto a natural need giving birth to a new mobile society.

Indeed, wireless networks found a strong success in the society as it deeply altered the social networks and communication habits. It also changed the telecommunication market plan, as in most of the countries, historical telecommunication companies experienced a constant decrease in the fixed line accounts to the benefit of wireless accounts. There was also a constant demand for more wireless services and communication capabilities, to which telecommunication companies responded by new generation networks from 2G, 2.5G, 3G, IMS or the forthcoming 4G. In the future, it is envision to be able to obtain similar services on hand-held devices as on standard desktop computers, including visio-calls and mobile television.

One major limitation of these wireless networks is the cost of building and maintaining the infrastructures. Without exception, those networks are based on a cellular approach, where all communications are routed by access points. The wireless medium providing much less bandwidth than its wired counterpart, wireless resource must be adequately managed. Moreover, the centralized aspect also generating bottlenecks at access points or base stations, very complex and ingenious user multiple wireless access had to be designed. One solution to reduce the cost of the infrastructure at the same time as improving network resource is to distribute routing tasks to the wireless terminals, giving birth to multi-hop wireless networks.

A MOTIVATIONS AND OBJECTIVES

Wireless Ad Hoc Networks are an extreme configuration of wireless networks, without a fixed or wired infrastructure, and where terminals are self-configuring in order to provide distributed multi-hop wireless communications. The lack of infrastructure or coordinator favors chaotic situations generating a large waste of already critical resources. Indeed, studies have shown that uncoordinated transmissions was not an efficient method to transmit data in wireless ad hoc networks as it creates an effect known as the "broadcast storm" problem. Similarly to a chaotic crowd, if everyone talks at the same time, no one can listen to anyone and everyone will have wasted its energy trying to communicate, no matter how loud they tried. This remark was one of the justification for the development of structures in ad hoc networks in order to improve data diffusion and energy consumption. Algorithms creating backbones and coordinations have yet to comply with two major assumptions: they must be *distributed* and *localized*. Indeed, ad hoc networks potentially being composed of a very large set of uncoordinated nodes, decisions should be taken at each node based on local information. As no omniscient coordinator exists in ad hoc networks, graph theory structures cannot be efficiently adapted. A successful new research area therefore appeared aimed at generating structures based on local information yet coming close to the optimal graph theory version: *Distributed Systems*. A wide range of solutions have been successfully developed to improve broadcast, transmission power or coordination.

A.1 The Burden of Mobility

Despite the efficient distributed solutions obtained in wireless ad hoc networks, a major assumption have been ignored: *mobility*. Advocates of distributed solutions argued that mobility could be simply seen as a maintenance duty, which is optimally kept local. Yet, this maintenance may also be seen as a waste of resource and a generation of instability and delays. Indeed, locality is not sufficient in order to efficiently maintain structures in wireless ad hoc networks, as mobility makes the structure adapted to past configurations, thus dooming them to inefficiency. Moreover, depending on the dynamics of the network, the local maintenance also becomes resource greedy. 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 significant energy and network resource dedicated to the maintenance of their routing tables.

Then years ago, the concept of *Kinetic Data Structure (KDS)* has been developed as a mean to efficiently adapt data structures to mobile objects and attributes. Observing that all then-known data structures were not directly applicable to a configuration of objects moving continuously, the objective was to benefit from the coherence and continuity in the motion of the points to gain efficiency. This could be achieved by the important hypothesis of objects knowing their trajectories and those of others. One possible application of KDS was to efficiently adapt spanning trees or other graph algorithms to mobile configurations. Beside the obvious requirement for the kinetic structure to fit the real topology, another performance measure was the locality of the structure's maintenance.

Yet, at that time, the major achievements in distributed computing obtained in recent years were not available. Accordingly, those two related problems with respect to ad hoc networks were handled separately. Indeed, *mobility* was studied for centralized graph algorithms, while *localized* graph algorithms were defined for static ad hoc networks. Observing that these two fields could be complementary, we propose here to regroup both assets in a new concept we named **Kinetic Graph**. Figure I-1 illustrates the two separate, yet complementary, issues of graph algorithms: central *vs.* local and static *vs.* dynamic methods.

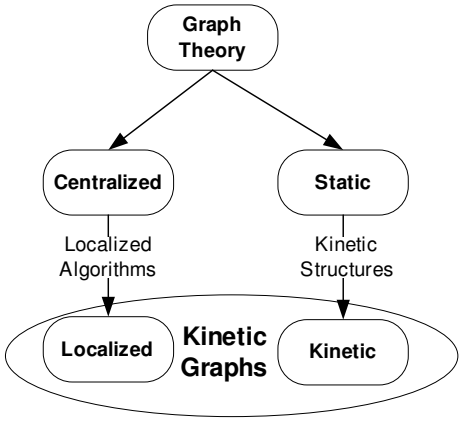


Fig. I-1. Illustration of the Positioning of Kinetic Graphs in Graph Theory

We therefore propose to borrow the localized management to the distributed research field and the kinetic management to the Kinetic Data Structure approach. We believe that *Kinetic Graphs* could be successfully applied to all approaches depending on structures, such as topology control, connected dominating sets, routing, or even location management.

A.2 The Mobility Reactive Approach

In the routing field of mobile ad hoc networks, protocols have been classified mostly in two classes, *Proactive* and *Reactive*, depending if a route is created if there is data traffic to transmit, or if all routes are proactively created independently of the data traffic.

In order to handle nodes mobility, all proactive and even reactive protocols send periodic beacon messages in order to detect any topological change and to adapt the routes. Taking a similar vocabulary, but considering mobility instead of routing, these protocols may therefore be seen as **Mobility Proactive**, in a sense that they proactively adapt to topology changes.

Conversely, a **Mobility Reactive** protocol anticipates all topological changes using mobility predictions. As long as any change has been correctly anticipated, mobility reactive protocols do not need to act and significant network resources may be saved. It is only when an unpredicted event occurs that this kind of protocols reacts, and it is the duty of the node that generated the unanticipated event to trigger the update.

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 protocols triggers a maintenance duty with or without change in the network topology, while the reactive approach triggers a maintenance duty if and only if there is an unanticipated topological change that effectively impacts the structures.

A very large majority of topology management, proactive or reactive protocols are Mobility Proactive. All mobility proactive approaches are resource demanding, as they need to periodically send beacon messages in order to detect a topology change. That also adds delays as the routing structure may only be updated when the protocol has obtained the new topology state.

Despite the growing interest in the mobility prediction protocols illustrated in Chapter II, all heuristics developed for Mobile Ad Hoc Networks did not aim at becoming mobility reactive. An intermediate class may then be defined: the **Mobility Adaptive** protocols. Instead of reactively adapting to topology changes, most of the approaches using mobility predictions try to reduce the periodicity of the proactive mobility maintenance by selecting overall optimal decisions. For instance, a node would choose the most stable node on average as its cluster head since this would reduce its maintenance duty. A tradeoff must then be considered between network optimality and maintenance cost.

To the best of our knowledge, the only technique which reaches the *Mobility Reactive* class and that can guarantee optimality at each time instance are Kinetic Data Structures (KDS) and **Kinetic Graphs**. A protocol based on Kinetic Graphs predicts the set of optimal decisions and dynamically switches from one to another at the right time. For example, a reactive routing protocol implementing the kinetic graphs will not only compute the optimal route from a source to a destination, but instead a *set* of optimal non overlapping routes, such that for each time instant, the optimal route is always available. Kinetic graphs are only reorganized when an unpredicted event occurs.

Before moving forward, we provide some necessary preliminary definitions related to graph theory. From *static* graph theory, we use the following definitions:

- **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

In the *kinetic* graph theory, we have basically the same definitions, but adapted to a moving structure:

- **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.

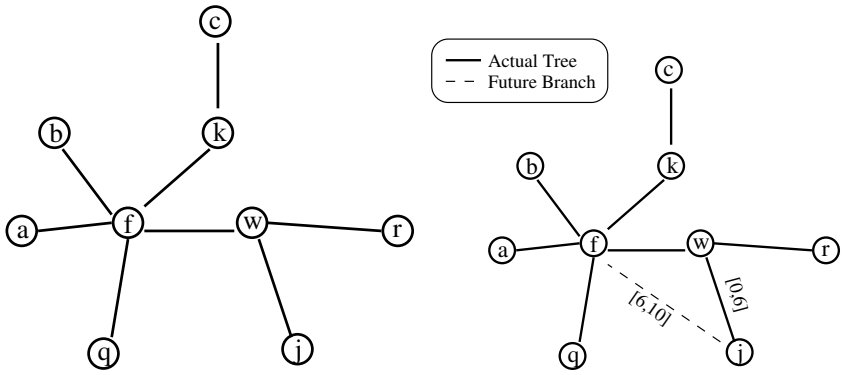
With the previous definitions and Fig. I-2, we are now ready to illustrate the difference between the traditional approach and kinetic graphs. Considering the distance between two nodes as the link weight, on Fig. I-2(a), an optimal tree is generated based on the criterion "shortest link weight between two nodes". However this three is only valid at precise moment when it is built, as mobility makes link weights and criteria change.

Without loss of generality, considering node j moving towards node f, there will be a moment when node j will be closer to node f than node w and therefore it should drop the link with node w to create a new link with node f. In order to detect this precise moment, the traditional approach is to periodically send beacon messages in order to check if the link is still the one with the best link weight. The kinetic approach instead compares the two time varying link weights and computes the transition. It then builds activations during which links always have the smallest link weight. It therefore does not need to periodically check this assumption, and thus does not need to send beacon messages, as long as the topology has been correctly anticipated.

This may be observed in Fig. I-2(b), where the transition has been found at time $t = 6s$ and where a link between node j and node w is always valid during the activation interval from $t = 0s$ to $t = 6s$, and a future link between node j and node f has been anticipated during the activation interval from $t = 6s$ and to $t = 10s$. Although the traditional approach is also able to detect the transition at time $t = 6s$ and update the tree accordingly, the kinetic approach does it without exchanging a single message as each node predicted the events.

The complexity of the Kinetic Graph approach and the need to construct not only the actual but also the predicted structures may be one reason for the very few related work in this specific field. To the best of our knowledge, Gentile and Haerri [1] were the first to introduce this idea to Wireless Sensor Network as a method to reduce the maintenance complexity of the Belman-Ford algorithm.

In this Thesis, we are going to describe different protocols in order to construct and maintain structures for Topology Control, Multipoint Relaying and for the Optimized Link State Routing (OLSR) protocol using Kinetic Graphs.



(a) Static Tree (b) Kinetic Tree

Fig. I-2. Example of the Kinetic Graph Approach

B CHALLENGES

The major assumption in Kinetic Graphs is the knowledge of nodes trajectories. However, in practice, this information is far from being straightforward to obtain and is, by itself, a complete and complex research area. The major challenge is therefore to be able to construct nodes estimated trajectories in order to use them for the Kinetic Graphs. For that matter, we will generate a prediction model and adapt it to fit mobility patterns created by a mobility model.

Previous to comprehensively describe the different aspects and relationships related to the trajectory knowledge, we first define four key concepts we will use in this thesis:

- **Trajectory–** It is defined as the probable course of a node in a mobile system.
- **Mobility Model–** It tries to model the "real" trajectories employed by nodes according to predefined configurations and external influence. A mobility model is mainly composed of a set of complex kinematic models aimed at realistically reproducing mobility patterns. Yet, realistic models also include external influences and correlations between mobile nodes and the topology.
- **Mobility Prediction–** A prediction model aims at estimating a probable trajectory modeled by a mobility model based on a partial knowledge of its mobility parameters. It is also composed of complex kinematic models and a criterion which must be correctly predicted.
- **Prediction Criterion–** A prediction criterion is the objective of the prediction and represents the information required by a particular application. It may be a distance, a nodal degree, a resource allocation, yet obtained based on the knowledge of nodes trajectories.

Those four concepts are closely interlinked in the management of mobile systems. Indeed, the efficiency of mobility prediction models come from their ability to extrapolate

the partially *unknown* mobility parameters and add them to known mobility parameters to estimate a trajectory generated by a mobility model. There is therefore a strong link between modeling and predicting mobility. In order to illustrate this interaction, we depict in Fig. I-3 a concept map of the interaction between those concepts in the case of a realistic mobility model.

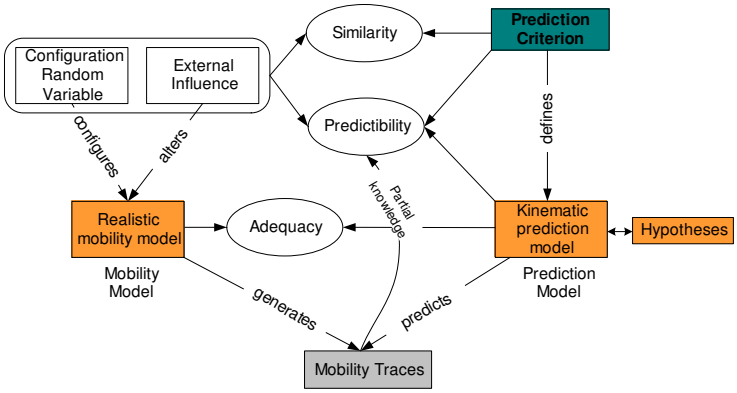


Fig. I-3. Illustration of the relationship between Realistic Mobility and Prediction Models

As the knowledge of the mobility model generating the pattern is not possible in practice, the first objective is usually to develop a kinematic model which fits best to real mobility patterns and which may be used to predict mobility. How close the synthetic kinematic model is from the real kinematic model defines the *adequacy*. The closer both are the better is the prediction. As the complexity of motion patterns exceeds the ability to develop a synthetic model, the problem is relaxed by using *hypothesis*. If these hypothesis are not valid, the prediction model will not be adequate with the mobility model. Nevertheless, a certain degree of error is necessary, and accepted in practice to model complex motion patterns. Finally, in order to further increase the complexity of the predictions, beside the unknown kinematic model used to generate motion patterns, a mobility model also employs random variable for its configuration and is altered by external actions. The prediction model may only obtain a partial knowledge of those variables.

B.1 Adequacy and Predictability

Before moving further in the analysis of the functionality of mobility and prediction models, we provides definitions of three key parameters describing their interactions.

Definition 1—Adequacy: The adequacy reflects the similarity between the motion models used by the kinematic model and the prediction model. If the two models are identical, or yield to identical results, one say that the two models are adequate.

Definition 2—Predictability: The predictability is a time interval between which the real and the predicted criteria are identical or within an acceptable error ϵ.

Definition 3—Similarity: The similarity is a feature only relevant to a mobility model

and reflects the relative variance 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 general, developing a mobility or a prediction kinematic model is a similar task and similar methods are used in order to generate kinematic models from real motion traces.

The predictability is only obtained either statistically or on average. 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 may only have a partial access to the parameters of the mobility model. 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 previous patterns, or statistically obtained if the motion patterns are modeled by an analytical model such as the RWM.

The similarity is obtained by the measure of the difference between previous values of the criterion. The similarity is minimized when no correlation exists between past and future values, while it is maximized when the future value may be fully extracted based on past ones.

We next illustrate the different parameters in the case of the Random Waypoint Model. It is acceptable for the prediction model to obtain the knowledge of the position and maybe the velocity, but neither the destination nor the epoch may be known. The hypothesis is a constant speed and azimuth between two trajectory changes, and the kinematic prediction model is a first order kinematic model $x = v \cdot t + x_0$. Accordingly, the model is in total adequacy with the RWM. By analyzing when the hypothesis are voided on the mobility traces, the predictability may be extracted, which is reduced to estimate the epoch time between two waypoints, and depends on the average speed and the size of the simulation area. Finally, the similarity is minimized as there is no correlation between past and future positions or speed.

B.2 Realism

As our objective is to adapt Kinetic Graphs on realistic configurations, the mobility patterns that are predicted should be as realistic as possible. The first step is therefore to model the mobility patterns really employed by nodes with a realistic mobility model.

Definition 4—Realism: The realism is the depiction of a feature as it appears in life, without error, interpretation or embellishment.

Yet, mobility models cannot be homogeneously realistic. A typical motion pattern should be targeted before being realistically modeled. Each application usually has its realistic mobility model, such as pedestrians or vehicular mobility.

In this thesis, we targeted the vehicular motions and therefore adapted the kinetic graphs to vehicular ad hoc networks (VANETs). Beside the applicability of this research field, there are strong reasons to be interested in vehicular networks. Although being usually sparse, at least in the sense of VANET-ready vehicles, and characterized by a very high node mobility, VANETs are however not energy limited and it is also widely accepted that vehicles may be equipped with GPS devices. This last hypothesis is very important in the case of Kinetic Graphs, as they need a mean to obtain geolocalization information and time synchronization to generate trajectories. Moreover, as we will show in this thesis, vehicular mobility patterns show non-uniform distributions of cars and velocity coming from a strongly restricted mobility helping to reduce the similarity error. Finally, the concept of trajectory may be easily seen in vehicular motions.

B.3 Prediction Errors

Once a prediction model is created, it is important to also estimate its prediction error. As this error depends on the criterion to be predicted, we therefore use the general term "Criterion Prediction Error", and define it with respect to the mobility patterns.

Definition 5—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 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 illustrate the criterion prediction error in Fig. I-4. 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. Then, the next step is to predict the modeled trajectory with a 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. If the criterion changes before the end of this interval, a predictability error ξ is generated, which is illustrated at time $t = 15$ on Fig. I-4. 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 Fig. I-4.

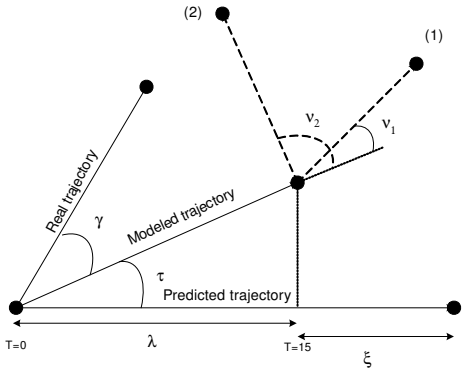


Fig. I-4. Illustration of criterion prediction error for the RWM model and a first order kinematic model

B.4 Prediction Strategies

In the previous sections, we introduced the concept of trajectory knowledge, described the relationship between mobility and prediction models, and provided a definition of the prediction error. We are now interested in analyzing this error depending on the prediction strategies.

- *Adaptive Strategy:* The node which generated the predictions corrects them at the end of the average predictability interval.
  - *Prediction Error:* $O(\gamma + \tau + \xi \cdot \upsilon)$.
- *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:* $O(\gamma + \tau)$.

The major difference between the adaptive and reactive methods comes from the node in charge for updating the predictions. While the former assigns this task to nodes making the predictions, the latter lets nodes being predicted update the system when necessary. Accordingly, the reactive approach is able to update the predicted trajectories at the exact predictability of each node, while the adaptative approach update at the average predictability interval.

The major benefit of the reactive approach is that the predictions are updated roughly at the same time as the mobility parameters, cancelling the predictability and the similarity errors. By adopting this approach, the prediction errors we still need to control are the adequacy and the realism. This is notably the solution employed by the *Kinetic Graphs*. In the case of the RWM model predicted by a first order kinetic model, we can reach an adequate trajectory construction and a kinetic graph maintenance as the adequacy is maximized and the predictability error cancelled. However, depending on the application, the realism error might be significant. We will further illustrate our statements in this Thesis.

In order to reduce the prediction errors while using the Kinetic Graphs, we therefore need to either reduce the realism or the adequacy errors. As the realism depends on the application, improving the realism may be done by using a specific mobility model for each application. However, this might generate trajectories that are so complex to model that we actually increase the adequacy error. One solution in order to have low realism and adequacy errors is to develop more sophisticated prediction models. Another solution is to reduce the sensitivity of the prediction criterion to prediction error.

As mentioned in Section B.2, the realism error is reduced in this Doctoral work by studying vehicular mobility and by using a realistic vehicular mobility model. A major limitation of Kinetic Graphs for vehicular networks is the complexity of vehicular motion modeling. Indeed, unlike random mobility models, motion patterns are influenced by external constraints that cannot be predicted. Accordingly, the motion and prediction models diverge and create a non negligeable adequacy error. We may then either improve the prediction model in order to better fit with vehicular motion patterns, or reduce the sensitivity of the prediction criterion to prediction errors. As the former is complex maybe even impossible as vehicular motions cannot be modelled solely by mathematical models, we chose the latter and we introduce a novel criterion called **Kinetic Nodal Degree**.

Indeed, a criterion based on the distance might not be stable enough with respect to prediction updates and errors in order to be efficiently used in Kinetic Graphs for Vehicular networks. The motivation for using nodal degree is manyfold. Indeed, successive criterion values are correlated and therefore limit the similarity error. Moreover, the predictability is also increased as the nodal degree does not change as frequently as the distance between two nodes. Accordingly, the adequacy error may be compensated by an increased predictability. We will illustrate this approach by adapting the kinetic nodal degree to the Multipoint Relaying (MPR) protocol and the OLSR routing protocol. We indeed obtain a significant improvement in both cases not only for the maintenance, which could be expected by a reactive maintenance, but also for the performance such as routing efficiency and delay.

B.5 Performance

As it may be seen in the definition of the prediction errors in Section B.4, the reactive strategy is naturally preferred as the prediction error is always smaller than the adaptive counter part. As Kinetic Graphs are natively implementing the reactive strategy, any protocol using Kinetic Graphs instead of the adaptive mobility maintenance strategy will have a better performance towards prediction errors.

Yet, one significant limitation of the reactive approach is the predictability interval λ. Indeed, as we update roughly at the instantaneous predictability interval, it becomes a performance criterion. If the predictability interval λ is short, we need to frequently update the Kinetic Graphs, while we may reduce this maintenance if λ is long. Summarizing this approach, Kinetic Graphs are characterized by

- Prediction Error: $O(\gamma + \tau)$.
- Performance: $O(\lambda)$

When the predictability interval is significantly reduced or too small to be efficiently used, the Kinetic Graphs fall to a degenerated case which is equivalent to a *Proactive Maintenance*, where the maintenance is periodically performed. Accordingly, Kinetic Graphs also cannot perform worst than mobility proactive protocols.

C CONTRIBUTIONS

Our contributions in the field of mobility modeling and prediction are manyfold. First, we defined the *Kinetic Graphs* approach, as an optimal solution for mobile wireless ad hoc networks that benefits from localized and distributed systems for improved graph algorithms, and kinetic structures for enhanced mobility maintenance. We described the neighborhood discovery phase, two different time varying link weights and an aperiodic neighborhood maintenance. This approach is in fact independent of the kind of link weights, but depends on the predictability of the motion patterns.

As the performance of Kinetic Graphs depends on the predictability interval, we then devised the predictability of two popular random mobility models, the Random Waypoint and the City Section mobility models. This first step was necessary in order to have a fair knowledge of the degree of predictability of those models, thus the expected benefit we could obtain by the use of prediction techniques as Kinetic Graphs.

Furthermore, as the prediction errors in Kinetic Graphs depend on the realism error, we studied realistic mobility models. The application we chose is vehicular networks as vehicles are more easily predictable and their mobility has a far more significant impact on routing protocols. Moreover, vehicular networks also provide solutions to obtain the localization information. Indeed, as for most mobility prediction schemes, Kinetic Graphs need a system that can provide accurate geographic positions. Then, as each node predicts future states based on the mobility of neighboring nodes, nodes synchronization is also required. One solution to solve both problems is the use of Geographic Positioning System (GPS) devices, which is a widely accepted assumption in Vehicular Networks. We therefore designed a realistic vehicular mobility model and validated it against a benchmark traffic generator.

Based on our findings on predictability and realism, we developed prediction models for mobility proactive routing protocols, as reducing the periodic maintenance of such proactive actions would be a major source of network improvement. We first developed a dynamic topology control protocol called KADER. However, due to the fact a trajectory change in any segment on the topology generated by KADER would actually impact the topology and make the system unstable, we then focused on a different approach in which we do not predict an exact trajectory, but instead a nodal degree. Indeed, even if the trajectory of a node changes too often to be accurately predicted, the nodal degree might not change that often.

We adapted our approach to the MPR protocol and designed the *Kinetic Multipoint Relaying(KMPR)* protocol, which is able to predict the actual and future Kinetic Multipoint Relays based on predicted nodal degrees. We compared our approach with MPR and illustrated how it was able to provide better broadcasting metrics and at a much reduced maintenance cost. We also tested this approach in conjunction with the OLSR protocol, which is based on the MPR protocol for the broadcast of its TC messages.

We also obtained significant improvements in routing metrics added with a much lower maintenance cost.

The results obtained with random mobility models are very motivating. However, it is also clear that the prediction model was able to bring significant improvement not only due to the prediction algorithm itself, but due to the lack of realism of the Random Waypoint model for instance. Indeed, vehicles have specific movements not well described by this model, and we chose to use a perfect adequacy at the cost of realism. We therefore also tested KMPR using realistic vehicular mobility patterns. Even though we expected the simplistic first order kinematic model used by KMPR not to be sufficient to accurately predict complex vehicular mobility, we illustrated that it was still enough to significantly improve the MPR and OLSR. Nevertheless, we expect more sophisticated prediction schemes to further improve our results.

D OUTLINE OF THESIS

This thesis focuses on studying and improving mobility management in ad hoc networks by using Kinetic Graphs. We also illustrate the strong required interaction between modeling and predicting mobility in order to efficiently adapt Kinetic Graphs to wireless ad hoc broadcasting and routing protocols.

Chapter II provides an overview of the current state of the art in mobility prediction models and their application to ad hoc networks. We also review current research orientations and achievements in the field of vehicular mobility modeling.

In Chapter III, we describe the *Kinetic Graph* approach, the hypothesis and methods used in order to successfully construct and maintain a graph in a mobile environment without requiring to a periodic maintenance. We then illustrate a potential application in Topology Control called KADER.

As mentioned in this introductory part, one of the main parameters and challenges for efficient mobility prediction is the analysis of the predictability. In Chapter IV we provide a lower bound for this parameter for the Random Waypoint Model (RWM). We show that this lower bound is more than sufficient to be beneficial to the Kinetic Graph approach.

Another important parameter also described in this Thesis is the realism. Choosing vehicular networks as the potential application, and noticing the lack of realistic vehicular mobility models, we introduce VanetMobiSim in Chapter V. It is a validated and freely available configurable traffic generator for vehicular ad hoc networks.

According to the particular mobility patterns generated by VanetMobiSim, we then study their impacts on the evaluation of ad hoc routing protocols in Chapter VI. We show how realism in vehicular motion modeling significantly alters routing in VANETs.

Chapter VII then provides an example of an application of the Kinetic Graph approach to broadcasting and routing in ad hoc networks. We developed the **Kinetic Multipoint Relays (KMPR)** and showed how kinetic graphs could significantly improve the maintenance overhead as well as the broadcast and routing efficiency either for random mobility models or the realistic VanetMobiSim model.

Finally, we draw concluding remarks and summaries of our work and contributions in Chapter VIII. We discuss the general lessons learned about mobility management in ad hoc networks, and outline some directions for future research in the different fields covered by this Doctoral Thesis.

CHAPTER II

Related Work

Contents

Abstract—*In this chapter, we first describe the challenges facing mobility prediction in mobile networks, and provide a related work on prediction models and their application to mobile ad hoc networks. We then illustrates the various issues and directions employed for modeling mobility, and provides a taxonomy of synthetic mobility models available for vehicular networks. We finally discuss the lessons learned from this dual analysis.*

A THE CHALLENGES OF PREDICTING MOBILITY

SINCE the creation of modern telephony, telecommunication networks have been developed in a static way, the mobility of users being negligeable with respect to the new capacity to connect two remote customers. At the eve of the Internet, we started to see the worldwide generalization of those static networks. However, at the same time, the customers demand for a larger flexibility toward nomadic patterns appeared, which placed mobility management within static networks as an important improvement factor.

In parallel, wireless networks were developed as the response of telecommunication operators for a growing demand of seamless mobility and wireless connections. Cellular networks such as 1G, 2G or 3G have been designed with a clear objective to offer a national mobile communication coverage, quickly increased to a worldwide coverage with the introduction of roaming capabilities. However, with the generalization of data traffic on cellular networks, as well as the increasing demand for improved throughput and security, mobility became a more serious subject.

Mobility is indeed a serious factor contributing to the performance of mobile telecommunication networks. It limits the capacity to maintain a connection, or to guaranty a quality of service between two customers. Moreover, the significant increase of the telecommunication customers dramatically increased the effect of mobility on the network maintenance.

Studies have been produced on the effect of mobility on telecommunication networks. Information Theory showed that mobility is able to increase the network capacity by increasing the network spatial diversity, a feature actually long known by epidemiologists studying virus propagations. However, this improvement comes at the cost of unbounded delays making this improved capacity unusable on real network deployments, and which explains in part why communication protocols are not taking advantage of the increased spatial diversity for communication improvements.

In most technologies used nowadays, networks are subject to terminal mobility. This effect may be compared to a blind person evolving in our universe and trying to discover its own representation with its stick. Our universe is indeed a knowledge plane acquired with experience, while mobile and fixed network stations are trying to blindly discover this universe using periodic transmission of beacon messages. Some protocols have been designed to reduce this drawback, yet without being able to jump the fence and resolve it.

In this section, we present concrete examples of methods successfully developed for telecommunication networks in order to limit the effect of user mobility. Then, in the second step, we re-introduce[1] a breach in the way modern telecommunication network could be designed by illustrating that mobility may actually be tamed by predicting instead of being subject to it. For that matter, we describe prediction models that are available, then we illustrate the application fields those models may be used. Our objective is to clearly demonstrate and convince the reader that efficient solutions exist

[1] We use the term "re-introducing", as this way of thinking had already been proposed in the past for the 2G and 2.5G cellular networks, but later forgotten in analysis of ad hoc and wireless telecommunication networks.

which provide a better vision of the mobility of network terminals.

A.1 Undergoing Mobility

As described in the previous section, telecommunication networks have always been subject to mobility. A major task for telecommunication engineers is therefore to design technics reducing this drawback. For example, in cellular networks GSM/UMTS, mobility management is handled differently whether the mobile terminal in active communication or not.

For example, the only way to contact a mobile terminal in idle mode is by paging it. In order to save network resources when the mobile terminal is not connected, the base station only keeps a coarse vision of the zone where the mobile terminal is. Accordingly, if the base station does not have any precise information on where a mobile terminal is, it needs to page the whole network with all the latency incurred by this method. Therefore, in order to reduce the drawbacks of this approach, GSM/UMTS systems developed a hierarchical structure called *Paging Area (PA)* including several *Location Areas* for the GSM or *Routing Areas* for the UMTS. Thanks to this, the system limits its paging's scope to the PA containing the last LA/RA where the mobile terminal has last been attached. By using this process, the system is able to save network resources and delay.

Instead, when the mobile terminal is actively communicating, the base station needs to keep a very precise vision on the region where the terminal is located in order to reserve network resources for future cell handovers. This procedure is critical for the seamless functionality of cellular networks as no drop calls should occur resulting from handovers. For that objective, the mobile terminal periodically samples then transmits RSSIs of all base station beacons it receives to its connected base station in order to obtain a coarse relative position estimate. Then, the mobile terminal and the attached base station may coordinate with the next base station in order to anticipate the handover and reserve the required network resources.

In the IP world, provisions have also been created in order to deal with mobility or nomadism of IP terminals. The IPv4 and IPv6 networks developed algorithms, called *Fast Handover*, limiting the packet losses generated by changes of covering zones. Alerts are triggered when a node is approaching a new access router, which creates alternate routes faster and re-routes packets even before the real hand-over actually takes place. However, this system is resource consuming as it requires a periodic tracking of access routers.

Ad hoc networks also had to quickly develop efficient methods to handle terminals mobility. Globally, five different categories of protocols were designed:

- **Proactive Protocols–** Similarly to static networks, those protocols build routing tables providing a path to any accessible destination on the network. Periodic beacon messages are triggered in order to adapt the backbone to topology changes at the cost of a higher energy consumption and channel occupancy. The two flagships in proactive routing protocols are the *Wireless Open Shortest Path First (W-OSPF) [2, 3]* and the *Optimized Link State Routing (OLSR) [4, 5].*

Indeed, after having developed many protocols, the community slowly started to converge to those two protocols, which are also the only two candidate to the IETF standard track RFC for proactive routing in Mobile Ad Hoc Networks (MANETs). Yet, the more the mobility increases, the harder it becomes to maintain the routing tables. Accordingly, this approach has shown not to be very adapted to fast mobile networks. Recent results also pointed out the relationship between performance and density, arguing that proactive routing could only be efficient on dense networks.

- **Reactive Protocols–** In order to limit the waste of resources, reactive networks only open routes on demand. Thanks to this limitation, the mobility of nodes not involved in the opened route does not influence network management. However, the mobility of nodes belonging to the opened route reduces the performance of reactive networks. Local repairs are possible in the case of a route failure and, in order to reduce the latency of a broken path, reactive networks also use periodic beacon messages. In this category, the *Direct Source Routing (DSR) [6]* protocol and the *Ad hoc On Demand Distance Vector Routing (AODV) [7]* are two potential candidate, although that the IETF recently chose a modified and improved version of AODV called *Dynamic MANET On-demand (DYMO) [8]* as the only candidate to IETF standard track RFC for reactive routing in MANETs.

- **Geographic Routing–** It is a stateless approach where no backbone or route is generated. Instead, geographic information of the destination and intermediate nodes are used in order to wisely choose the best candidate to forward a packet toward the intended destination. Those protocols are based on two functions: *the greedy forwarding* and the *recovery*. Indeed, each node receiving a packet will try to chose the best candidate among its neighbors with the maximum progress toward the destination node. This is the *greedy forwarding* phase and *Most Forward within Radius [9]* is the technique most widely used in order to find the best progress. But in some cases, the packet falls in some local maxima, where not any single node in the neighborhood may bring any potential progress toward the destination. Accordingly, a recovery phase is triggered, where the packet is sent back until an alternate candidate is found. This is the *recovery phase* and use mostly *Face Routing ([10] pp. 389-394)* to circumvent the local maxima. The first and still pioneer protocol in this field is the *Greedy Perimeter Stateless Routing (GPSR) [11]* protocol, but some extension and improvements in the two phases have been suggested ([10] Sect. 12.4). Nodes mobility still alters the precision of geo-localization information, potentially reducing the performance of the geographic forwarding approach. The strong requirement of the availability of a geo-localization system was the major justification for the IETF for not pushing this approach for standardization. Yet, the stateless feature of geographic routing made them good candidates for routing in Vehicular Ad Hoc Networks, where GPS systems are commonly accepted.

- **Fish-Eye Routing–** In order to deal with the lack of precision of geographic information, mobility is handled in a different way whether the destination node is far or close from the intermediate or sender node.

  - *Locally*: Frequent position updates of all neighboring nodes are triggers as mobility has a significant local influence.

  - *Remote*: Only a coarse mobility maintenance is triggered as the remote mobility does not have a significant influence on a local decision.

  The *Fisheye State Routing (FSR) [12]* protocol or the *Landmark Routing (LANMAR) [12]*, two proactive approaches, are two examples, where a node keeps up to date state information about all nodes in its inner circle (or landmark), while the accuracy of such information decreases as the distance increases. Even if a node does not have an accurate state information about distant nodes, packets will be routed correctly because the route information becomes more and more accurate as the packet gets closer to the destination. Another proactive protocol in this category is called *Distance Routing Effect Algorithm for Mobility (DREAM)*. It is based on *location information*, and adapts its location updates to both mobility rate and distance. Finally, a reactive approach called *Location-Aided Routing (LAR) [12]*, also based on *location information*, has been developed, where each node maintains the location about nodes it is aware of with respect of the distance. The farer is the node, the larger is area and then, on demand, orients route requests toward the area where the destination node is.

- **Hybrid Routing–** This is the last category of protocols which mixes the proactive approach for local routing and reactive even geographic approach for distance routing. Most of the protocols developed in this category either create local zones, clusters, or trees and use a reactive routing strategy to route between them. The *Zone Routing Protocol [13]*, or the Hybrid Ad Hoc Routing Protocol [14] are examples of this approach.

Although some techniques have been developed to reduce the impact of nodes mobility, it still has a major impact on the performance of routing protocols. And similarly to the topology management approaches previously described, all are subject to mobility, and non negligeable resources are dedicated to maintaining the stability of the network backbone or routes with respect to mobility. These resources could be better used if mobility could be used as an asset instead of a drawback.

A.2 Predicting Mobility

An alternative to the methods described in the previous section is to try to predict users mobility. Indeed, by again considering the example of blind persons, what differentiate us from them is first our global long range vision, and second our capacity to predict and anticipate the evolution of our environment. Similarly, mobility prediction techniques could be used in order to improve the management of mobile networks.

Definition 1—Mobility Prediction: Capacity to evaluate a future position given past positions.

Mobility Prediction is actually a very ancient technique used by the first sailors to navigate on seas and oceans. In marine literature, this technique is better known as **Dead Reckoning**. Using instruments measuring:

- the initial point

- the azimuth, or headings (Astrolab, Sextant, Compas)
- the speed (Chip log, Tachometer, Anemometer, Doppler sonar)
- the time (Astrolab, Chronograph)

the dead-reckoning technique is able to obtain the current position and the distance travelled since the last known position. Inertial systems are able to improve the precision of dead-reckoning techniques for systems that are not able to receive satellite signals. Nowadays, a large variety of navigational methods are still based on dead-reckoning, varying from under-water navigation, spatial navigation, missile guidance and tracking. More generally, in any domain where a knowledge of the trajectory taken by a system is vital, mobility predictions are used.

In telecommunication network management, resources are shared in order to benefit to the widest set of users. And those resources are allocated depending on the density of users. Yet, mobility makes this management random and inefficient. The knowledge of the trajectory taken by users may be very useful in order to improve the resource management of mobile telecommunication networks. This is also a significant motivation for the study of mobility prediction techniques in the field of telecommunication networks.

Available Localization Techniques

On periodic position reassessments, mobile terminals using algorithms based on mobility prediction techniques must acquire their position. It is therefore necessary to obtain a sporadic access to a geo-localization system. Three categories of geo-localization algorithms exists:

- Satellite Systems (GPS or Galileo): It is a widely diffused system, which guaranties a precise localization ($\pm$ 1m) at a low cost. However, the acquisition time may be long ($>$30s), it also consumes a non negligeable energy, and requires access to satellite signals.
- Beacon Systems (GSM): The precursor of GPS localization, and an alternative to situation when the GPS signal is not available. However, the precision cannot challenge the GPS system.
- Hybrid Systems:
  - Inertial Systems: Contain a set of accelerometers, gyroscopes and guidance algorithms able to provide the velocity, orientation, and angular velocity of a mobile system by measuring the linear and angular accelerations applied to the system in an inertial reference frame. If calibrated on known positions and velocities, the inertial system is then able to estimate a mobile system complete trajectory.
  - GPS Systems: Compute position, velocity, acceleration, and use complex mobility prediction techniques when the signal is not available.

– GSM Systems: Uses all available techniques or triangulation or multilateration in cellular networks: Angle of Arrival (AOA), Time Difference of Arrival (TDOA), Enhanced Cell Identification (E-CID), Uplink Time Difference of Arrival (E-TDOA), Enhanced Observed Time Difference (E-OTD), or A-GPS.

Mobility Prediction Models

Mobility prediction models has nothing new, and working on this field could look like trying to reinvent the wheel. Indeed, they have initially been developed for tracking purposes in the 60s and in cellular Networks since 1995. More complex models have later been used in order to be applied to cellular systems requiring a quality of service such as the Wireless ATM network. The complexity and the precision of those models culminated around the year 2001, but unfortunately were forgotten afterward as illustrated in Fig. II-1[2]. In fact, it took a long time to the mobile ad hoc network community to understand that mobility predictions were as important as it used to be for cellular network. But then, only simplistic models were re-introduced, as if the whole past literature has either been forgotten, or judged too complex for the needs. However, we started to see a growing popularity in complex iterative models in recent works and expect this popularity to further increase in the near future. This section aims at recalling and putting back into the light the different orientations taken by the community in order to tame the mobility of mobile terminals.

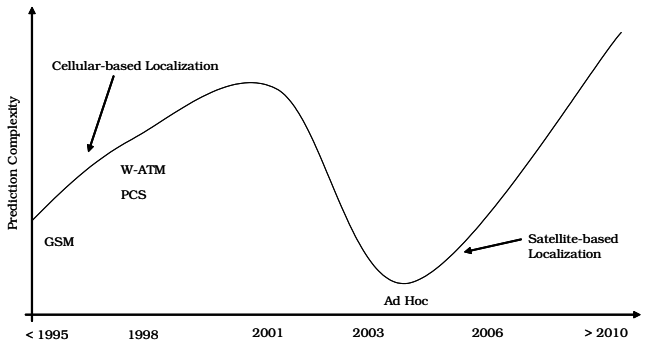


Fig. II-1. Evolution of the Popularity of Prediction Techniques

Deterministic Models

Deterministic mobility prediction models may be a first order model, only considering the position and a fixed velocity, but more higher order models have also been designed, including acceleration and a time-varying velocity.

The mostly known and used deterministic model is the first order kinetic model illus-

[2]This figure has been obtained based on the number of google hits using keywords *Trajectory, Prediction, Tracking, Mobility* and estimated based on the complexity of each solution and the initial publication year.

trated in Fig. II-2(a).

$$\overrightarrow{Pos}(i+1) = \begin{pmatrix} X_{i+1} \\ Y_{i+1} \end{pmatrix} = \begin{pmatrix} X_i \\ Y_i \end{pmatrix} + \begin{pmatrix} V_x^i \\ V_y^i \end{pmatrix} \cdot (t - t_i) \tag{II-1}$$

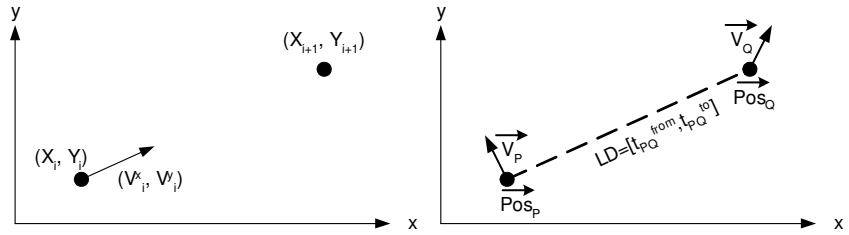
A direct application of (II-1) is to compute the kinetic distance or the estimated connection time between two nodes P and Q. The kinetic distance is computed as follows

$$\begin{aligned} D_{PQ}^2(t) &= D_{QP}^2(t) = \|\overrightarrow{Pos}_Q(t) - \overrightarrow{Pos}_P(t)\|_2^2 \\ &= \left[\begin{pmatrix} X_Q - X_P \\ Y_Q - Y_P \end{pmatrix} + \begin{pmatrix} V_Q^x - V_P^x \\ V_Q^y - V_P^y \end{pmatrix} \cdot t \right]^2 \\ &= a_{PQ}t^2 + b_{PQ}t + c_{PQ}, \end{aligned} \tag{II-2}$$

Considering r as nodes maximum transmission range, as long as $D_{PQ}^2(t) \leq r^2$, nodes P and Q are neighbors. Therefore, solving

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

gives t_{PQ}^{from} and t_{PQ}^{to} as the time intervals during which nodes i and j remain neighbors (see Fig II-2(b)).



(a) First Order Prediction Model

(b) Link Duration (LD) between nodes P and Q

Fig. II-2. First Order Prediction Model and its Application to Link Duration

In cases where the velocity is not constant, a second order prediction model based on the Euler motion law is used.

$$\overrightarrow{V}_{i+1} = \overrightarrow{a_i} \cdot t + \overrightarrow{V}_i \tag{II-4a}$$

$$\overrightarrow{Pos}_{i+1} = \frac{1}{2}\overrightarrow{a_i} \cdot t^2 + \overrightarrow{V}_i \cdot t + \overrightarrow{Pos}_i \tag{II-4b}$$

Although vehicular motions involve impulsive forces (such as sudden braking), a constant acceleration is usually accepted in high speed mobile networks. However, a piecewise constant acceleration is used in practice. In both cases, (II-4) may be used to

predict a future position based on some kinetic information. (II-4) may be solved by substitution.

$$\begin{aligned}\overrightarrow{Pos}_{i+1} &= \frac{1}{2}\cdot\frac{\left(\vec{V}_{i+1}-\vec{V}_i\right)}{t}\cdot t^2+\vec{V}_i\cdot t+\overrightarrow{Pos}_i \\ &= \frac{\left(\vec{V}_{i+1}+\vec{V}_i\right)}{2}\cdot t+\overrightarrow{Pos}_i \end{aligned} \tag{II-5a}$$

Accordingly, position predictions are calculated using a velocity one step ahead, which forces us to have two samples of past velocities and two piecewise constant accelerations in order to predict the future position.

$$\vec{V}_i = \vec{V}_{i-1}+\vec{a}^{(i-1)\to i}\cdot(t_i-t_{i-1}) \tag{II-6a}$$

$$\vec{V}_{i+1} = \vec{V}_i+\vec{a}^{i\to(i+1)}\cdot(t_{i+1}-t_i) \tag{II-6b}$$

$$\overrightarrow{Pos}_{i+1} = \frac{\left(\vec{V}_{i+1}+\vec{V}_i\right)}{2}\cdot(t_{i+1}-t_i)+\overrightarrow{Pos}_i \tag{II-6c}$$

where $\vec{a}^{(i-1)\to i}$ and $\vec{a}^{i\to(i+1)}$ are the constant acceleration during the time intervals $[t_{i-1},t_i]$ and $[t_i,t_{i+1}]$ respectively.

If the acceleration is constant between two sampling intervals, the velocity increases linearly with time and the approximation $\frac{\|\vec{V}_{i+1}\|+\|\vec{V}_i\|}{2}\approx\|\ \vec{V}_{i+\frac{1}{2}}\ \|$ is exact. If not, we need to sample the velocity at the mid-interval and use a variation from II-4 called the *Feynman-Verlet* model. The *leap-frog* algorithm may be appropriately used in this case.

$$\vec{V}_{i+\frac{1}{2}} = \vec{a}_i\cdot t_i+\vec{V}_{i-\frac{1}{2}} \tag{II-7a}$$

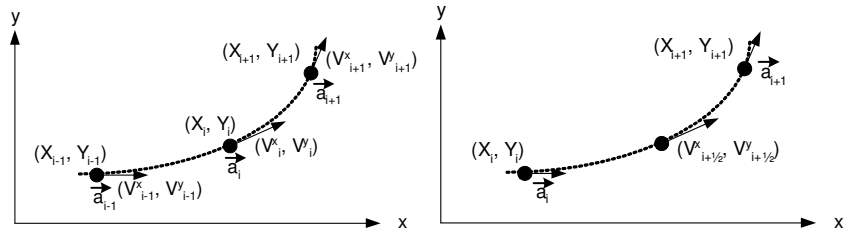
$$\overrightarrow{Pos}_{i+1} = \vec{V}_{i+\frac{1}{2}}\cdot t+\overrightarrow{Pos}_i \tag{II-7b}$$

Changes in position are calculated using a velocity that is half a step ahead in time. Likewise, changes in velocity are calculated using an acceleration which is half a step ahead in time. Position and acceleration are therefore in-phase, while velocity is out of phase with position and acceleration.

The *leap-frog* algorithm owes its simplicity to the fact that stepping the velocity half step out of phase with the position and acceleration provides midpoint values for both (II-7a) and (II-7b) and thus provides more accurate results than the Euler model. Fig. II-3 illustrates both approaches.

Stochastic Models

Stochastic models do not aim at obtaining an exact prediction, but rather a correct one with high probability. Stochastic models may be easily used to add an uncertainty to deterministic predictions. But a more important use is to model unknown parameters in



(a) Euler Model (b) Feynman Model

Fig. II-3. Second Order Constant Acceleration Prediction Models

the state equations or to take into account the model's prediction error. In many cases, it is both. For example, tracking-based auto-regressive processes (AR) use white noise to model the AR prediction errors. The estimation of the states, such as position or velocity, is often accomplished using Kalman Filters. Even if position or velocity are obtained without error, the AR process still provides predictions with some errors. Now, if errors are added to the positions or velocities, the performance may decrease drastically. Accordingly, in most applications, joint optimization is applied to obtain good predictions. In the rest of the section, we provides examples and related work using stochastic prediction models.

Approaches for mobility tracking mostly rely on *Autoregressive processes* [15, 16], *Kalman Filtering* [17, 18, 19, 20], *semi-Hidden Markov* [21, 22, 23] models, or *Particle Filtering* [24, 25, 26, 27]. Two measurements have been mostly used in the literature, the Received Signal Strength Indicator (RSSI) or the Time or Arrival (TOA), but GPS positioning is experiencing a growing interest from the community as a mean to reduce the measurement error.

The first and most straightforward model is to weight a deterministic prediction by the probability the prediction still exists. It is defined as follows

$$Pred^{stoch}(t) = Pred^{det}(t) \cdot e^{-\beta(t-t_{sample})} \tag{II-8}$$

where

$Pred^{det}(t)$	:	Deterministic Mobility Prediction at time t
$e^{-\beta(t-t_{sample})}$	:	Stochastic validity of the prediction parameters
β	:	Stability of the mobility parameters (also called Predicability)
t_{sample}	:	Latest sampling time of the mobility parameters

As mentioned in the beginning of this section, *Autoregressive (AR) Models* also falls in the stochastic class. A white Gaussian noise with zero mean ϵ is used to model the AR prediction error, and the mobility state of the process is estimated using Kalman or Particle Filters.

An auto-regressive model of order p defines the n^{th} value as a weighted sum of the p previously measured ones and is defined as

$$x_n = \alpha_0 + \sum_{i=1}^{p} \alpha_i x_{n-p} + \epsilon_n \tag{II-9}$$

where ϵ_n is an independent identically distributed noise with zero mean.

Creixell and Sezaki [15] proposed to model pedestrian trajectories using first order auto-regressive process (AR(1)) for the velocity and the azimuth. They used Least Square Lattice filters (LSL) to solve their model and obtained fairly good predictions up to 10 simulation step ahead.

Zaidi and Mark [16] also used a first order auto-regressive model (AR(1)) but used Yule-Walker formulation in order to estimate α and error coefficients of the AR(1) process. Unlike [15], the mobility state are not obtained from mobility traces, but are measured using RSSI (Received Signal Strength Indicator) or TOA (Time of Arrival) and then approximated using Kalman Filters. They validated their approach by comparing it against real sets of data.

Another model is called the *Gauss-Markov* prediction model and has been proposed in [28]. It first models a node's velocity as a time-correlated Gauss-Markov random process. In discrete time, it computes the predicted velocity based on the previous value and a Gaussien iid process

$$v(n) = \alpha v_{n-1} + (1-\alpha)\mu + \sigma\sqrt{1-\alpha^2} w_{n-1} \tag{II-10}$$

with the Gauss-markovien auto-correlation process

$$R_v(\tau) = E\left[v(t)v(t+\tau)\right] = \sigma e^{-\beta|\tau|} + \mu^2 \tag{II-11}$$

where

α	$=$	$e^{-\beta\|\tau\|}$
β	:	Memory size
σ^2	:	Variance of the v(t) process
μ	:	Expectation of the v(t) process
w_n	:	Gaussien IID Process

Their numerical results have demonstrated the importance of the performance gain of prediction-based approaches, but also confirmed that the performance of such approach is directly proportional to the predictability of a node's mobility pattern. That was also our intuition and was the justification of our predictability analysis in Chapter IV.

While deterministic models are able to model quite fairly first order or second order kinetic models with constant accelerations, a velocity subject to an unknown acceleration, or known but non-constant acceleration requires the use of more complex stochastic models. Many studies of position tracking in wireless networks exist, most of them deploying some form of Kalman filtering to the position tracking problem. However, in recent years, it has been noted that the sequential Monte Carlo processing filters, better known as the Particle Filter, can provide an improved performance in the non-linear and non-Gaussian noise tracking problem [29].

Central to all navigation and tracking applications is the motion model to which various kind of model based on filters can be applied. Models that are linear in the state dynamics and non linear in the measurements are often considered:

$$x_{t+1} = Ax_t + B_u u_t + B_f f_t \quad \text{(II-12a)}$$
$$y_t = h(x_t) + e_t \quad \text{(II-12b)}$$

where

x_t : state vector
u_t : measured input
f_t : unmeasurable input or faults toward the measured input
y_t : measurement
e_t : measurement error

An independent distributions may usually be assumed for f_t, e_t, x_0 with known probability densities p_{e_t}, p_{f_t}, and p_{x_0}, respectively, not necessarily Gaussian. The difference between the applications based on (II-12) mainly lies in the different means to obtain the measurement equation (II-12b).

Liu and al. [17] proposed a mobility model for wireless ATM networks based on a dynamic linear system model in wich the mobility state consists of the position, velocity and acceleration of the mobile terminal. Originally introduced by Singer [30], the system can capture a wide range of realistic user mobility patterns. The measurement is based on an estimated position obtained by the RSSI (received signal strength indicator) from three different base stations, and the state vector is given by

$$\dot{x}(t) = Ax(t) + Bu(t) + Cr(t) \quad \text{(II-13)}$$

where

$$A = \begin{bmatrix} \Theta & 0 \\ 0 & \Theta \end{bmatrix} \qquad B = C = \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix}$$
$$\Theta = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \Phi = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and where

$x(t) = \left[X(t)\dot{X}(t)Y(t)\dot{Y}(t)\right]^T$: Node u mobility vector
$u(t) = [u_x(t)u_y(t)]^T$: Node u deterministic acceleration command
$r(t) = [r_x(t)r_y(t)]^T$: Node u random acceleration

The structure of the model, illustrated in Fig. II-4, manages to replace a time varying acceleration with a semi-Markov based acceleration commands and a random acceleration component. This filter is resolved using Kalman Filtering techniques.

A key observation is that the process u_n is a semi-Markov process. Therefore, accurate estimation of u_n should exploit its semi-Markov characterization. Yu and Kobayashi [21,

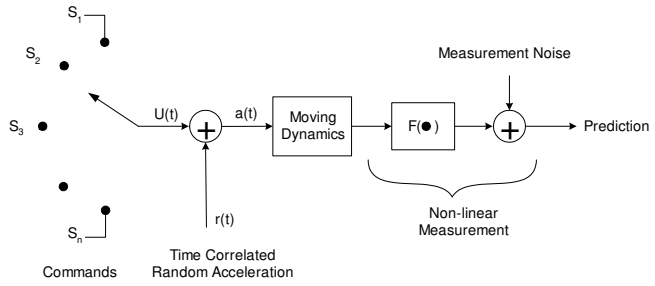


Fig. II-4. Structure of the Stochastic Micro-Prediction Model

22] developed an efficient algorithm for estimating the parameters of a hidden semi-Markov Model (HSMM), a generalization of a similar approach for Hidden Markov Models (HMM), and its application to position tracking. Zaidi and Mark [23] also used the same idea to obtain the acceleration command u_n with a HSMM estimator while using a Kalman Filter to estimate the mobility states. Thanks to this hybrid approach, their solution outperformed Liu's work [17] in terms of prediction errors by a factor of 5.

Zaidi et al. [20] later generalized their approach and proposed to first preprocess the RSSI with an average Filter to obtain coarse position estimates, and second to decouple the mobility state estimates $x(t)$ from the estimation of the discrete command process $u(t)$. They illustrated how their approach was able to follow mobile trajectories more accurately than in Liu's work.

Pathirana et al. [18] proposed a modification to Liu's work and used a Robust Extended Kalman Filter (REKF) approach in order to improve the prediction accuracy, processing efficiency, and more important, to include non-linearities to the model. Accordingly, no assumption is made on the measurement equation or the system dynamics, and thus could be able to better model sharp turns and log-normal or Nakagami propagation models popular in the modeling of Vehicular Network.

Another mean to solve the motion model described in (II-12), without using REKF or more complex systems, is by means of Bayesian recursive filtering, also called Particle Filtering. The optimal Bayesian Filter in the case of (II-12) is given below and is composed of a prediction step and an update step. If the set of available observations at time t is given by

$$Y_t = \{y_0, \cdots, y_t\},$$

then the Bayesian solution to compute the posterior distribution $p\left(x_{t+1}|Y_t\right)$ of the state vector, given past observations, is given by

$$
\begin{aligned}
p\left(x_{t+1}|Y_t\right) &= \int p\left(x_{t+1}|x_t, Y_t\right) p\left(x_t|Y_t\right) && \text{(II-16a)}\\
&= \int p\left(x_{t+1}|x_t\right) p\left(x_t|Y_t\right) dx_t && \text{(II-16b)}\\
&= \int p_{f_t}\left(B_f^{\dagger}\left(x_{t+1} - Ax_t - B_u u_t\right)\right) p\left(x_t|Y_t\right) dx_t && \text{(II-16c)}\\
p\left(x_t|Y_t\right) &= \frac{p\left(y_t|x_t\right) p\left(x_t|Y_{t-1}\right)}{p\left(y_t|Y_{t-1}\right)} && \text{(II-16d)}
\end{aligned}
$$

where we assume that both the initial probability density of state p_0, and the density $p\left(x_t|Y_t\right)$ at time step t are known and $p\left(y_t|Y_{t-1}\right) \approx c_t$.

In the case the motion model is as II-12a and the update equation is as II-12b, (II-16) may be rewritten

$$
\begin{aligned}
p\left(x_{t+1}|Y_t\right) &= \int p_{f_t}\left(B_f^{\dagger}\left(x_{t+1} - Ax_t - B_u u_t\right)\right) p\left(x_t|Y_t\right) dx_t && \text{(II-17a)}\\
p\left(x_t|Y_t\right) &= \frac{p_{e_t}\left(y_t - h(x_t)\right) p\left(x_t|Y_{t-1}\right)}{c_t} && \text{(II-17b)}
\end{aligned}
$$

The particle filter can be considered as an approximation to a sequential solution to the above equations. It achieves this by representing the posterior density with some random weighted samples, called the *Particles*. A typical particle filter algorithm consists of 5 steps that we shortly describe next.

1. *Initialization*: Generate $x_0^i \sim p_{x_0}, i = 1, \cdots, N$. Each sample of the state vector is referred to as a *particle*.

2. *Measurement Update*: At each particle position, the assigned weight of each particle is updated and normalized according to a likelihood function (based for example on RSSI cumulative distribution).

$$w_t^i = w_{t-1}^1 p(y_t|x_t^i)$$

$$w_i = \frac{w_i}{\sum_i w_t^i}$$

   This is the *update step* of the Bayesian recursive filtering.

3. *Resampling*: P particles are replaced from the set of particles based on the weights. This step is necessary in order to avoid a high concentration of probability mass at a few particles.

4. *Prediction* Move the particle forward according to the adopted Model (II-12a for instance). This step is therefore the *prediction step* of the Bayesian recursive filtering. The particles are now referred to as *predicted particles*.

5. Let $t := t = 1$ and iterate item 2).

There is a large literature of successful use of Particle Filtering methods to solve positioning, tracking or navigation, and it is hard to be exhaustive. The major difference between different approaches are usually the *measurement step* or the *Resampling Phase*.

For example, Yang and Wang [24] also illustrated the inaccuracy of Liu's work and proposed an alternative estimation scheme based on a sequential Monte Carlo (SMC) Filtering. The SMC can achieve better performance than the Liu's filtering, but is computationally intensive and hence might not be suited for real-time trajectory predictions. Zaidi et al. [20] later showed that the SMC was outperformed by their Modified Kalman Filtering approach.

Gustafsson et al. [25] proposed a framework for positioning, navigation and tracking problems using particle filters. They showed a clear improvement in performance in real-time, off-line, on real data and in simulation environments compared with existing Kalman filter-based solutions in term of convergence time and precision. By using Rao-Blackwellization, authors also managed to reduce the increased computational complexity of the Particle filter approach compared to Kalman Filters.

Mihaylova et al. [26] presented two other Sequential Monte Carlo algorithms, a Particle Filter and a Rao-Blackwellised Particle Filter. In contrast to previous work [17, 20, 24], the mobility tracking is formulated as an estimation problem of a *hybrid system*, where a base state vector is continuously evolving, and where a mode state vector may undergo abrupt changes. This formulation together with the Monte Carlo approach showed it could reduce the computational complexity and provide efficient mobility tracking.

Sha et al. [27] described another Particle Filtering approach for position tracking in Wi-Fi networks under the assumption of log-normal fading and with intermittent GPS information signaling. They obtained a factor 2 improvement against a stand alone Wi-Fi-based localization and could obtain real-time positioning in hybrid Wi-Fi and GPS systems.

History-based Models

Those models are usually used to predict the terminal macro-mobility, or the cell to cell mobility. Indeed, repetition of routine movements allows to more easily learn the users preferred paths.

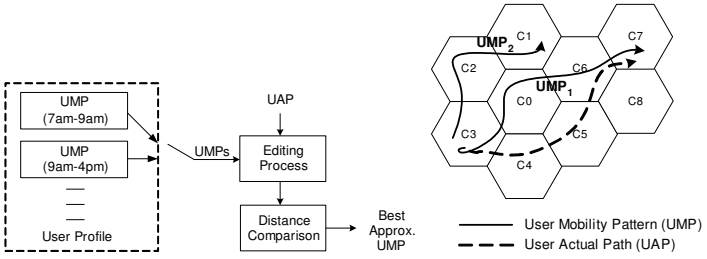
One method to characterize the regularities of users mobility is to record a set of *User Mobility Patterns* stored in a *profile* for each user and indexed by the occurrence time (see Fig. II-5(a)). The major difficulty is to assess the sensitivity between the UMP and the *User Actual Path (UAP)*. Indeed, is a UAP, which diverges from the UMP by a single cell, a small variation of the same path or a totally new path not reported in the profile ?

Different approaches have been proposed [31, 17, 32, 33]. We illustrate here the solution presented in [17], where the authors successfully used approximate pattern-matching techniques to find the UMP that fits best to a UAP. For example, if a UMP is described by a cell sequence $(a_1 a_2 \cdots a_{i-1} a_i a_{i+1} \cdots a_n)$, then the authors modelled the regular movement of a mobile user as an edited UMP by allowing the following legal options:

- *inserting* a cell c at position i of the UMP gives UAP:$(a_1 a_2, \cdots a_{i-1} c a_i a_{i+1} \cdots a_n)$
- *deleting* the cell a_i at position i of the UMP gives UAP: $(a_1 a_2, \cdots a_{i-1} a_{i+1} \cdots a_n)$
- *changing* a cell a_i to another cell c gives UAP: $(a_1 a_2, \cdots a_{i-1} c a_{i+1} \cdots a_n)$

Figure II-5(b) gives an example a UMP $c_3 c_0 c_6 c_7$ and its edited UAP $c_3 c_4 c_5 c_6 c_7$, which can be obtained by *changing* c_0 to c_4 and *inserting* c_5.

The degree of resemblance of a UAP with a UMP is measured by the *edit distance* a finite string comparison metric. The simplest way to find this distance is by determining the smallest number of *insertion*, *deletion* and *changes* by which two cell sequences can be made alike. If the *edit distance* is less than a matching threshold t, an approximately matched UMP is found, indicating the general moving intention of the user and the macro-prediction may be done accordingly. For example, in Fig. II-5(b), the UMP_1 has clearly a smaller *edit distance* than UMP_2 compared to the UAP. UMP_1 is therefore selected as general moving intention of the user.



(a) Macro-model

(b) Example with 2 Potential UMPs

Fig. II-5. Global Mobility Model

In the next example, another way to benefit from the repetition of mobility patterns is by modeling them by a sequence of stationary events generated using a *Markovien process of order m*. In other words, the new event may be generated as a function of the *m* previous events.

$$P\left[V_{l+m+1} = v_{l+m+1} | V_1 = v_1 \cdots V_l = v_l \cdots V_{l+m} = v_{l+m}\right] \quad \text{(II-18a)}$$

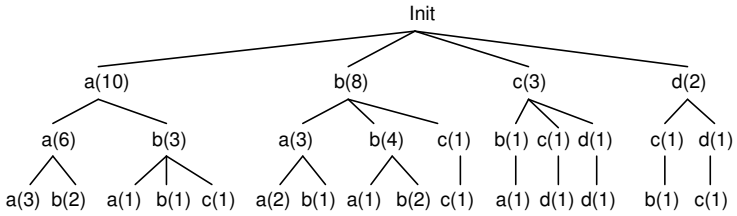
$$= P\left[V_{l+m+1} = v_{l+m+1} | V_l = v_l \cdots V_{l+m} = v_{l+m}\right] \quad \text{(II-18b)}$$

where V_i are the states of the system, which may be represented by a cell, or an occupied road segment.

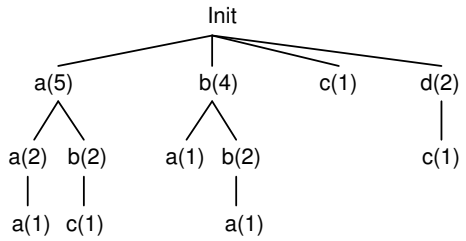
A representation of (II-18) may be obtained by a *trie* or a *digital search tree*, where every node represents a context $V_k = v_k | V_{k-1} = v_{k-1} \cdots V_1 = v_1$ and stores its last symbol along with the relative frequency of its appearance at the context of the parent nodes. Obviously, the depth of the trie is the order of the Markovien process and, as we move down the trie, we restrict our uncertainty to finally converge to a next event prediction when we reach a leaf. The performance of the prediction is therefore the trie's ability to add a new event to the frequency of an already existing node (reducing the uncertainty) and not to create a new branch (an unpredicted event).

The *Uncertainty* of a new event based on a sequence of past events is called the *Entropy* in Information Theory, and the optimal prediction of the future state may then be obtained from algorithms minimizing this entropy. The Lempel-Zif (LZ78) algorithm is a good choice in order to generate an optimal dictionary of observed paths and a reduced search trie.

Figure II-6 illustrates an example of the trie representation of a movement history $aaababb\ bbbaabccddcbaaaa$ with a second order Markov Process and its improvement using the LZ78 algorithm. This algorithm creates the dictionary a, aa, b, ab, bb, bba, abc, c, d, dc, ba, aaa and only adds a new branch by concatenating a new entry v with a symbol w already contained in it.



(a) Classical Trie



(b) Lempel-Zif Trie

Fig. II-6. Trie Representation of a Movement History modeled by a Markovie Process of order 2

The idea of using the LZ78 algorithm in order to reduce the uncertainty has been originally presented by Bhattacharya and Das [34] for Location Management under the name *LeZi-Update*. An extension to Handover prediction in Wireless Networks, which has been introduced in [35], is presented next.

By representing the state sequences as $N, H_1, H_2, \cdots H_n, E$, where N is a new call, H_i is the i^{th} handoff, and E is the end of call, we can generate the prediction tree illustrated in Fig. II-7. Each sequence of events $N, H_1, H_2, \cdots H_n, E$ during the lifetime of a call corresponds to a substring in the Ziv-Lempel algorithm.

Each node builds a tree based on the sequence of events. For example, in Fig. II-7, the Lempel-Zif algorithm found 3 substrings which ended at time slot 2, 3 substrings which contained a handover to cell b also at time slot 2, and where all of them ended

right after the handover, or 15 substrings which contained a handover at time slot 1 etc...

When the mobile requests a new call in cell a in the time interval 9:00-9:01 a.m., we can use the statistics preserved in the node's mobility trie to predict the probabilities of the next possible events of this mobile. From the root's point of view, it will terminate the call without handoffs in the 2nd time slot with probability of $3/56$, handoff to cell b in the 2nd time slot with probability of $2/56$. Then, depending on the next event, we go down the tree following the sequence of events in order to refine the predictions. If one prediction error occurs, the tree is updated.

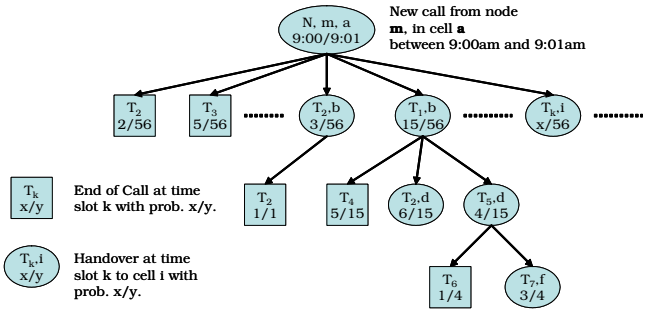


Fig. II-7. Example of a Lempel Zif tree predicting the time slot of either the end of call or a handoff

This kind of repetition also allows to successfully use fuzzy logic algorithms. In the following example, authors in [36] used a Neuro-Fuzzy Inference Model (NFIS), which is based on an `IF`, `THEN` rule whose consequence is a real number.This model provides the inference structure that avoids the time-consuming process of defuzzification in an inference procedure. The form of fuzzy `IF-THEN` rules is as follows:

$$Rule\ i: \ \ IF\ x_1 = A_1^i\ and\ \cdots x_j = A_j^i,\ \ THEN\ y = \omega_i \tag{II-19}$$

where

$$\begin{aligned} x_1 \cdots x_d &: \text{input variable} \\ A_j^i &: \text{A fuzzy set for input variable } x_j \text{ in the } i^{th} \text{ fuzzy rule} \\ \omega_j &: \text{A real number for output variable in the } i^{th} \text{ fuzzy rule} \end{aligned}$$

Given the real-value input vector $\vec{x} = [x_1, x_2 \cdots x_d]$, the real-value output of the fuzzy model is inferred as follows:

$$NFIS = f\left(\vec{x} = [x_1, x_2 \cdots x_d]\right) = \frac{\sum_{i=0}^{n} \mu_i \omega_i}{\sum_{i=0}^{n} \mu_i} \tag{II-20}$$

where

$$\mu_i = \prod_{j=1}^{d} \mu_{A_j^i}(x_j) \quad : \quad \text{Fuzzy membership function of the Fuzzy set } A_j^i$$

$$\mu_{A_j^i} \quad : \quad \text{Fuzzification function (triangular, trapezoid, Gaussian)}$$

Then, the prediction is as follows:

$$NFIS\left(S_k^t, S_k^t, \cdots S_k^{t-(d-1)}\right) \rightarrow S_k^{t+1} \tag{II-21}$$

where S_k^t is the state k a time t. S_k^t may contains a set of parameters such as velocity, acceleration or azimuth.

In [37], the sectorization-based prediction model has been proposed which has been applied to cellular networks. It is mostly a refinement from the basic regular path prediction model where a next cell is predicted based on a sequence of previous visited cells. Depending on predefined sectors in a cell, a node will be more likely to move the cell adjacent to its sector, or move to another sector. Fig. II-8 illustrates the sectors in a cellular network.

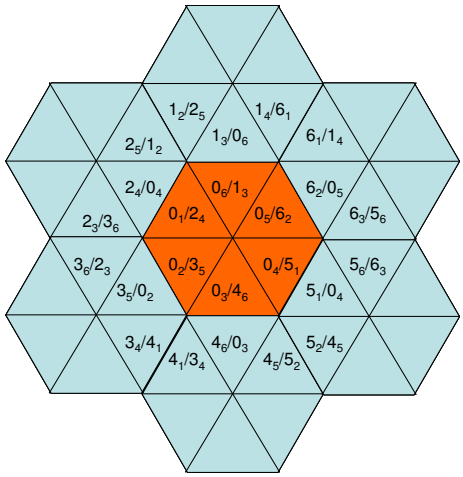


Fig. II-8. The Cell Sector Numbering Schema

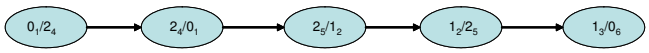


Fig. II-9. Sectorized Mobility

Based on the cell sector numbering schema, a history-based sectorized mobility is generated as illustrated in Fig. II-9, where the probability to be at position m after X movements is given by

$$P_X(m) = \frac{X! p^{\frac{1}{2}(X+m)} (1-p)^{\frac{1}{2}(X-m)}}{\left[\frac{1}{2}(X+m)\right]! \left[\frac{1}{2}(X-m)\right]!} \tag{II-22}$$

where p is the probability to leave the cell.

Another approach introduced in [38] is the *Shadow Cluster* model, which is also a refinement from the basic regular path prediction model. Its concept is that any active wireless device establishes an influence on cells in the vicinity of its location and its direction of travel. The cells currently being influenced are said to form a *Shadow Cluster* because the region of influence follows the movement of the active device like a shadow. Fig. II-10 illustrates this approach, where the shaded areas compose the shadow cluster centered in cell C.

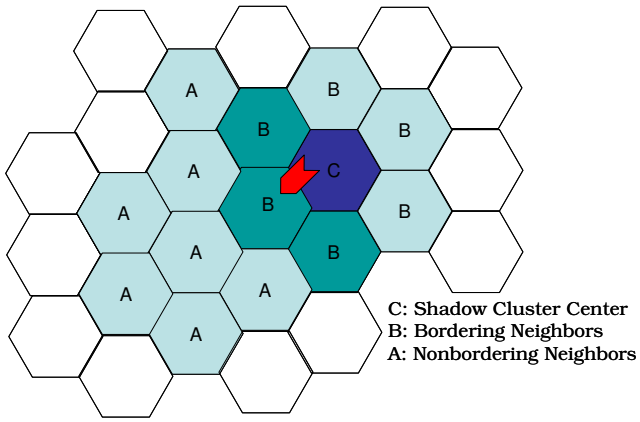


Fig. II-10. Shadow Clusters Produced by an active mobile terminal

Complex stochastic techniques are used to compute the active mobile probabilities to generate the shadow clusters. In [39], Akyldiz and Wang further improved the *Shadow Cluster* approach to consider aggregate history and a stochastic model of cell residence time to shrink the region considered for shadow clusters.

Finally, another particular class of history-based model uses Neural Networks. In those models, information is gathered in order to train the neural network, which then is able to predict a particular future state of the network. Based on the sequence of input vectors during the training period, back-propagation is used to update and improve the weights of the neural networks layers. Depending on the needed complexity, such neural network may have several hidden layers. Fig. II-11 illustrates an example of the multi-later neural network with back-propagation.

where

X_i : State of node i
Y : Decision (Handover, Link duration)
Z : Real Value during training
δ_i : Weight correction during training

In Capka and Boutaba [40], the moving trajectory of a mobile node is determined as a sequence of base stations the node has been attached to. The neural network is trained with sequences observed in the past in order to detect the current movement pattern and improve network management.

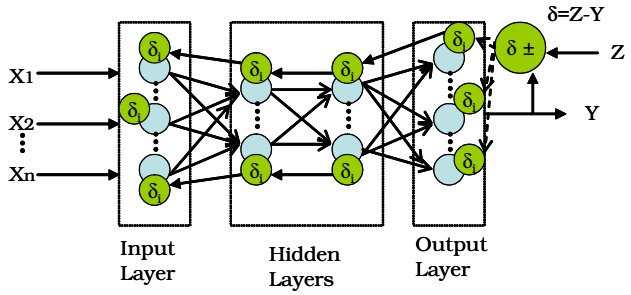


Fig. II-11. A Multi-later Neural Network with Back-propagation

Shang et al. [41] developed a clustering-based protocol using wavelet neural network, where a wavelet function replaces the Sigmoid function in conventional neural networks. They showed that this approach resulted in more stable clusters than LowID or MaxConn[3].

Hierarchic Models

This last category includes the most precise models ever developed at this time. Indeed, hierarchic models usually include a micro-prediction algorithm coupled with a macro-prediction schema. Most of the time, we find stochastic models for the micro-prediction, and history-based models for the macro-prediction. Those models are not only capable of predicting with a very high precision the sequence of cells a user will use in the future, but also the time it will reach the limit of each cell.

A.3 Network Algorithms using Prediction Models

In this section, we describe the divers application domains where mobility prediction schemes have been successfully adapted to mobile ad hoc networks.

Figure II-12 illustrates domains where prediction models could be applied in mobile ad hoc networks. In most of those domains, protocols have been developed which significantly improved the network performance.

Mobility prediction techniques have been successfully applied to the following domains:

- Connection Management:
  - KADER [42]: This protocol generates a connected forest using a non-periodic maintenance strategy. Its Performance is similar to other topology control algorithms, yet at a drastic reduction of the maintenance overhead.
  - Kinetic MultiPoint Relays (KMPR) [43]: The KMPR protocol elects MPR nodes depending on their predicted nodal degree. The KMPR protocol is

[3]LowID is a clustering technique where a node with the lowest ID is elected as cluster head, while MaxConn elects a node with the maximum connectivity as clusterhead

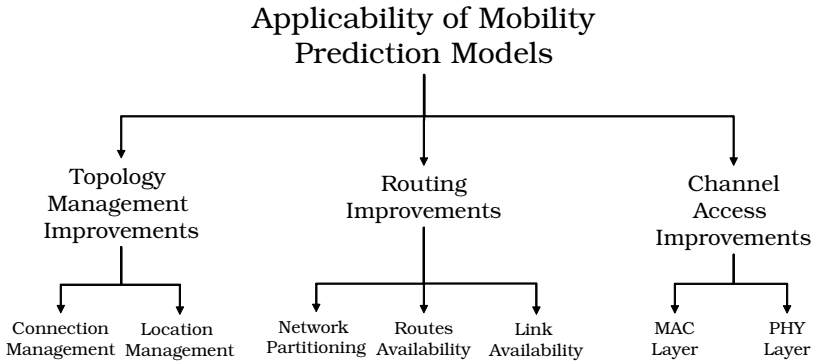


Fig. II-12. Classification of the Applicability of Prediction Techniques

able to reduce the MPR protocol maintenance overhead by 60% and the delay by 25

- Location Management:
  - Dead Reckoning-Based Location Service [44]: This model adjusts the periodic dissemination of geographic information based on a first order deterministic mobility prediction model.
  - Mobility Prediction-based GLS [45]: Improves the Grid Location Server (GLS) by adapting the periodic location maintenance with two prediction models *deterministic first order* and *history-based first order Markovien*.
  - Predictive Location Service (PLS) [46]: This approach only uses a first order deterministic prediction model, but manages to reduce the location errors and the maintenance overhead of GLS.
- Link Availability:
  - Mobility Prediction-based Position-based Forwarding (MP-PBF) [47]: This approach improves the accuracy of the general PBF protocol by relaying packets depending the predicted position of the intermediate nodes with respect to the predicted position of the destination.
  - Predictive Location Aided Routing (P-LAR) [48] : This approach sectorizes nodes mobility. The cost of routes establishment is largely lower than LAR.
  - Prediction-based Link Availability (PB-LA) [49]: Uses a deterministic first order prediction model to efficiently predict the link duration between two nodes. When used in conjunction with the DSR protocol, the performance is significantly improved.
  - Context Aware Routing (CAR) [50]: Uses a Stochastic model based on Kalman Filters in order to predict mobility in sporadically connected networks. It illustrated its benefit compared with traditional epidemic routing.

- Route Availability:

  - Distance Vector with Mobility Prediction (DV-MP) [51]: Represents the link cost as the predicted link duration for Distance-Vector approaches, and improves their performance with respect to conventional Distance-Vector protocols or LAR.

  - Kinetic Minimum Spanning Trees (KMST) [1]: KMST uses a stochastic prediction model in order to build a Spanning Tree using a non-periodic maintenance strategy.

  - Dead-Reckoning Model (DRM) [52]: The DRM improves the performance of DSR by using the prediction of links duration instead of the hop count as the cost metric.

  - Reliable On-Demand Routing Protocol (RORP) [53]: This approach opens routes with weight set to the minimum link duration of each link comprised in the route. Then, the source chooses the route based on the maximum link duration.

  - AODV MOvement Prediction Routing (AODV-MOPR) [54]: In AODV-MOPR, nodes selected to establish a route from a source to a destination node are selected depending on their similar direction and velocity. This generates a 7% improvement in AODV route stability.

  - Kinetic Link State Routing (KLSR) [55]: Used in conjunction with KMPR, KLSR is able to use the actual and predicted future MPR selectors in order to build an optimal routing table containing not only actual but also future optimal paths, without requiring the periodic broadcast of Topology Control (TC) messages.

A.4 Summary

Telecommunication networks have long been subject to the effect of user mobility. In order to reduce this drawback on cellular networks (PCS, GSM), mobility prediction models have been created and successfully tested. Among other examples, this approach was seen as a way to provide some kind of quality of service to Wireless ATM networks. However, the prediction approach lost its popularity as those telecommunication networks were replaced by new systems such as 3G or wi-fi networks.

The subject reclaimed its popularity when mobility became again a major source of waste of network resource. For instance, predicting mobility in mobile ad hoc networks is seen as a mean to make such system scalable.

A large literature reading illustrates the advantage of mobility prediction models for mobile ad hoc network. However, unlike their counterpart in cellular networks, almost none of them uses complex schemes. At the eve of mesh and vehicular networks, it would be interesting to reintroduce such approach. Moreover, the effect of prediction models on the physical or the Mac layer have not been studied yet.

B MOBILITY MODELING

Vehicular Ad-hoc Networks (VANETs) have been recently attracting an increasing attention from both research and industry communities. One of the challenges posed by the study of VANETs is the definition of a generic mobility model providing an accurate, realistic vehicular mobility description at both macroscopic and microscopic levels. Today, most mobility models for vehicular studies only consider a limited macro-mobility, involving restricted vehicles movements, while little or no attention is paid to micro-mobility and its interaction with the macro-mobility counterpart. On the other hand, the research community cannot have access to realistic traffic generators which have not been designed to collaborate with network simulators. In this paper, we first introduce a classification of existing methods for the generation of vehicular mobility models, then we describe the various approaches used by the community for realistic VANET simulations. Finally, we provide an overview and comparison of a large range of mobility models proposed for vehicular ad hoc networks.

B.1 A Framework for Realistic Vehicular Mobility Models

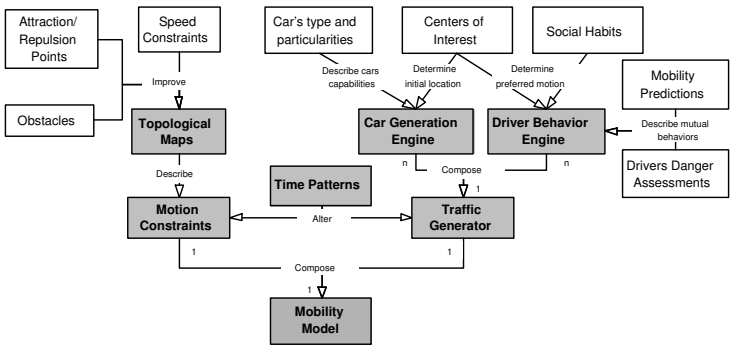


Fig. II-13. Proposed concept map of mobility model generation for inter-vehicle communications

In the literature, vehicular mobility models are usually classified as either **microscopic** or **macroscopic**. When focusing on a macroscopic point of view, motion constraints such as roads, streets, crossroads, and traffic lights are considered. Also, the generation of vehicular traffic such as traffic density, traffic flows, and initial vehicle distributions are defined. The microscopic approach, instead, focuses on the movement of *each* individual vehicle and on the vehicle behavior with respect to others.

Yet, this micro-macro approach is more a way to analyze a mobility model than a formal description. Another way to look at mobility models is to identify two functional blocks: **Motion Constraints** and **Traffic Generator**. **Motion Constraints** describe how each vehicle moves (its relative degree of freedom), and is usually obtained from a topological map. Macroscopically, motion constraints are streets or buildings, but microscopically, constraints are modeled by neighboring cars, pedestrians, or by limited roads diversities either due to the type of cars or to drivers' habits. The **Traffic Generator**, on the other hand, generates different kinds of cars, and deals with their

interactions according to the environment under study. Macroscopically, it models traffic densities or traffic flows, while microscopically, it deals with properties like inter-distances between cars, acceleration or braking.

The framework states that a realistic mobility model should include:

- **Accurate and Realistic topological maps:** Such maps should manage different densities of roads, contains multiple lanes, different categories of streets and associated velocities.
- **Smooth deceleration and acceleration:** Since vehicles do not abruptly break and move, deceleration and acceleration models should be considered.
- **Obstacles:** We require obstacles in the large sense of the term, including both mobility and wireless communication obstacles.
- **Attraction points:** As any driver knows, initial and final destination are anything but random. And most of the time, drivers are all driving in similar final destinations, which creates bottlenecks. So macroscopically speaking, drivers move between a repulsion point towards an attraction point using a driver's preferred path.
- **Simulation time:** Traffic density is not uniformly spread around the day. An heterogeneous traffic density is always observed at some peak time of days, such as *Rush hours* or *Special Events*.
- **Non-random distribution of vehicles:** As it can be observed in real life, cars initial positions cannot be uniformly distributed in a simulation area, even between attraction points. Actually, depending of the *Time* configuration, the density of cars at particular *centers of interest*, such as homes, offices, shopping malls are preferred.
- **Intelligent Driving Patterns:** Drivers interact with their environments, not only with respect to static obstacles, but also to dynamic obstacles, such as neighboring cars and pedestrians. Accordingly, the mobility model should control vehicles mutual interactions such as overtaking, traffic jam, preferred paths, or preventive action when confronted to pedestrians.

The approach can be graphically illustrated by a concept map for vehicular mobility models, as depicted in Figure II-13.

B.2 Generating Mobility Models for Vehicular Networks

Although being a promising approach, the proposed Framework in the previous section suffers from non negligeable drawbacks and limitations. Indeed, parameters defining the different major classes such as *Topological Maps*, *Car Generation Engine*, or *Driver Behavior Engine* cannot be randomly chosen, but must reflect realistic configurations. Therefore, due to the large complexity of such project, the research community took more simplistic directions and moved step by step.

Globally, the development of modern vehicular mobility models may be classified in four different classes: *Synthetic Models* wrapping all models based on mathematical models, *Traffic Simulators-based Models*, where the vehicular mobility models are extracted from a detailed traffic simulator, *Survey-based Models* extracting mobility patterns from surveys, and finally *Trace-based Models*, which generate mobility patterns from real mobility traces. A proposed classification is illustrated in Fig. II-14

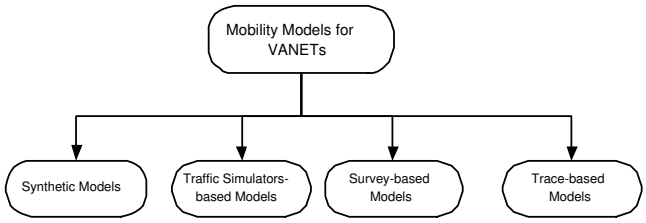


Fig. II-14. Classification of Vehicular Mobility Models

Synthetic Models

The first and most well known class includes the synthetic models. Indeed, major studies have been undertaken in order to develop mathematical models reflecting a realistic physical effect. Fiore wrote a complete survey of models falling into this category. We shortly summarize the basic classification he developed. For a more complete version, we refer the reader to [56]. According to Fiore's classification, Synthetic models may be separated in five classes: *stochastic models* wrapping all models containing purely random motions, *traffic stream models* looking at vehicular mobility as hydrodynamic phenomenon, *Car Following Models*, where the behavior of each driver is modeled according to vehicles ahead, *Queue Models* which model roads as FIFO queues and cars as clients, and *Behavioral Models* where each movement is determined by a behavioral rule imposed by social influences for instance. Fig. II-15 illustrate Fiore's classification.

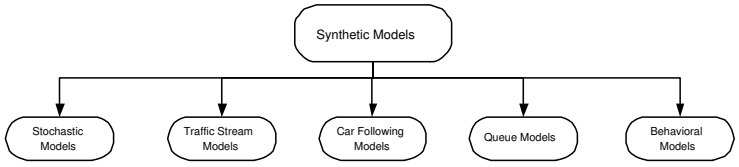


Fig. II-15. Classification of Synthetic Mobility Models

In order to validate a mathematical model, it should be compared to real mobility. Accordingly, one solution is to gather mobility traces by large measurement campaigns then compare the patterns with those developed by the synthetic model. In [57], authors proposed to validate some key characteristics of the RWP such as average speed and rest times using real life data. The Weighted Waypoint Model (WWM) [58] is a second attempt to validate a synthetic model which has been tuned by real traces. The WWM adds the notion of preference to the random waypoint. This calibration of this preference criterion has been performed based on mobility traces obtained inside

the USC campus. The HWGui [59] generates realistic time dependant highway traffic patterns that have been validated against real traffic in German Highways.

A major critique from synthetic models is the lack of realism towards human behavior. Indeed, drivers are far from being machines and cannot be programmed for a specific behavior, but respond to stimuli and local perturbations which may have a global effect on traffic modeling. Accordingly, realistic mobility modeling must also consider behavioral theory, social networks for instance, which makes models far from being random. Musolesi and Mascolo illustrated this approach in [60] by developing a synthetic mobility model based on social network theory, then validating it using real traces. They showed that the model was a good approximation of human movement patterns.

Survey-based Models

Although the behavioral theory is able to generate macro-motion models or deviation from micro-motion models, another solution is a calibration by means of comparison with realistic social behavior. The major large scale available surveys come from the US department of Labor, which established surveys and gathered extensive statistics of US workers' behaviors, going from the commuting time, lunch time, traveling distance, preferred lunch politics and so forth. By including such kind of statistics into a mobility model, one is able to develop a generic mobility model able to reproduce the non random behavior observed in real daily life urban traffic.

The UDel Mobility Model [61] falls into this category. Indeed, the mobility simulator is based on surveys from a number of research areas including time-use studies performed by the US Department of Labor and Statistics, time-use studies by the business research community, pedestrians and vehicle mobility studies by the urban planning and traffic engineering communities. Based on these works, the mobility simulator simulates arrival times at work, lunch times, breaks/errands, pedestrian and vehicular dynamics (e.g., realistic speed-distance relationship and passing dynamics), and workday time-use such as meeting size, frequency, and duration. Vehicle traffic is derived from vehicle traffic data collected by state and local governments and models vehicle dynamics and diurnal street usage. We can also cite the Agenda-based [62] mobility model, which combines both the social activities and the geographic movements. The movement of each node is based on individual agenda, which includes all kind of activities on a specific day. Data from the US National Household Travel Survey has been used to obtain activity distribution, occupation distribution and dwell time distribution.

Trace-based Models

Another major drawback of synthetic models is that only some very complex models are able to come close to a realistic modeling of vehicular motion patterns. A different approach has therefore been undertaken. Indeed, instead of developing complex models, and then validating them using mobility traces or surveys, a crucial time could be saved by directly extracting generic mobility patterns from movement traces. Such approach recently became increasingly popular as mobility traces started being gathered through various measurement campaigns launched by projects such as Craw-

DaD [63], ETH MMTS [64], UMASSDieselNet [65], MIT Reality Mining [66], or USC MobiLib [67]. The most difficult part in this approach is to extrapolate patterns not observed directly by traces. By using complex mathematical models, it is possible to predict mobility patterns not reported in the traces to some extends. Yet, the limitation is often linked to the class of the measurement campaign. For instance, if motion traces have been gathered for bus systems, an extrapolated model cannot be applied to traffic of private vehicles.

Another limitation from the creation of trace-based mobility models is the few freely available vehicular traces. Several research groups are currently implementing testbeds, but the outcome might only be available in few months or years if they are even made available to the public. To corner this issue, some teams (ETH [64]), or the Los Alamos Research Labdeveloped very complex simulators, which are able to generate very realistic vehicular traces. However, due to the complexity of the simulator, the trace generation time has an order of magnitude of couples of hours or days. Then, this mobility data are usually considered as real traces for the generation or calibration of mobility models.

Tuduce and Gross in [68] present a mobility model based on real data from the campus wireless LAN at ETH in Zurich. They used a simulation area divided into squares and derive the probability of transitions between adjacent squares from the data of the access points. In [69], authors combine coarse-grained wireless traces, i.e., association data between WiFi users and access points, with an actual map of the space over which the traces were collected in order to generates a probabilistic mobility model representative of real movements. They derived a discrete time Markov Chain which not only considers the current location, but also the previous location and the origin and the destination of the path. However, this study does not consider correlation between nodes.

Kotz et al. [70] describe a measurement technique for extracting user mobility characteristics also from coarse-grained wireless traces. They derived the location of users over time and also emphasize popular regions. They major findings was unlike standard synthetic mobility models, the speed and the pause times follow a log-normal distribution. They also confirmed that the direction of movement closely reflects the direction of roads and walkways, and thus cannot be modeled by a uniform distribution. Similarly to [68], they ignore correlation between adjacent nodes.

In [71], user mobility is modeled by a semi-Markov process with a Markov Renewal Process associated with access point connection time instants. Unlike previous studies, this work is able to model how user mobility is correlated in time at different time scales. The authors also performed a transient analysis of the semi-markov process and extracted a timed location prediction algorithm which is able to accurately predict users future locations. This work is moreover the first attempt to characterize the correlation between movements of individual users.

Chaintreau et al. [72] studied the inter-contact time between wireless devices carried by humans using coarse-grained wireless traces but also experimental testbeds using iMotes. Their major breakthrough was that unlike the widely accepted assumption that inter-contact time follows an exponential distribution, a more realistic assumption should be that the distribution exhibit a heavy tail similar to a power law distribution. Another study ([73]) analyzed the student contact patterns in an university campus us-

ing class time-tables and student class attendance data. A major restricted assumption has been taken, which forces students to either be in classrooms or in some randomly chosen communication hubs. They showed that in this configuration, most students experienced inter-contact time of the order of magnitude of few hours. However, unlike other studies (such as [72]), the inter-contact time does not follow a power law distribution. This is where the limitation from trace-base mobility modeling appears. Indeed, this study is specific to class attendance, and results obtained remain also specific to the environment where the study has been made.

By using traces, various research teams have therefore been able to extract mobility models that would reflect more realistically to motion we experience in real life. Moreover, a major result from trace-based mobility modeling, which is at odd with hypothesis used by synthetic models, is that the speed and pause time distributions followed a log-normal distribution, and that the inter-contact time may be modeled by a power law distribution,

Traffic Simulator-based Models

By refining the synthetic models and going through an intense validation process using real traces or behavior surveys, some companies or research teams gave birth to realistic traffic simulators. Developed for urban traffic engineering, fine grained simulators such as PARAMICS [74], CORSIM [75], VISSIM [76] or TRANSIMS [77], are able to model urban microscopic traffic, energy consumption or even pollution or noise level monitoring. However, those simulators cannot be used straightaway for network simulators, as no interface have been developed and traces are mutually incompatible. This, added to the commercial nature of those traffic simulators, became the "raison d'être" for the development of the novel off-the-shelf vehicular mobility models that we are going to describe in this Chapter.

By developing parser between traffic simulator traces and network simulator input files, the end-user gains access to validated traffic patterns and is able to obtain a level of detail never reached by any actual vehicular mobility model. The major drawback of this approach is the configuration complexity of those traffic simulators. Indeed, the calibration usually includes tweaking a large set of parameters. Moreover, the level of detail required for vehicular network simulator may not be as demanding as that for traffic analysis, as global vehicular mobility patterns and not the exact vehicular behavior are, by far, sufficient in most cases. We also want to emphasize that those commercial models require the purchase of a license that may exceed thousands of dollars, which is a major limitation for the VANET research community.

B.3 Mobility Models and Network Simulators: The Mute talking to the Deaf

In the previous section, we described various approaches that have been undertaken by the research community in order to develop realistic mobility models adapted to vehicular traffic. Yet, in order to be used by the networking community, those models need to be made available to network simulators. And this is precisely where we fall into a "kafkaien" situation. The worlds of Mobility Models and Network Simula-

tors may be compared to a mute talking to a deaf. They have never been created to communicate, and even worse, they have been designed to be controlled separately, with no interaction whatsoever. When imagining the promising applications that could be obtained from vehicular networks, where networks could alter mobility, and where mobility would improve network capacity, this situation cannot be tolerated anymore if the vehicular networks community has the means of its ambitions.

Initially, mobility was seen by network simulators as random perturbations from optimum static configurations. Then, in order to give some control to the user on the mobility patterns, network simulators became able to load mobility scenarios. There is virtually no limitation to the complexity of the models handled by those simulators, loading scenarios extracted from traffic simulators, or complex synthetic models for instance. However, as illustrated in Fig. II-16, the models must be generated prior to the simulation and must be parsed by the simulator according to a predefined trace format. Then, no modification of the mobility scenario is allowed.

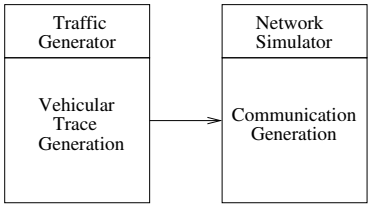


Fig. II-16. Interaction between Network and Traffic Simulators: The Isolated Case

For example, VanetMobiSim [78] is able to generate realistic vehicular mobility traces in urban area as well as highway scenarios. It models car-to-car interactions and car-to-infrastructure interactions, which allows it to integrate stop signs, traffic lights, safe inter-distance management and behavior based macro-mobility. It is also able to generate mobility incidents such as accidents. Moreover, it is freely available and has been validated against realistic traces obtained from CORSIM, a validated traffic generator.

Beside the general waste of computational resources, no interaction is therefore possible between those two worlds. Unfortunately, all historical models and most of the recent realistic mobility models available to the research community fall into this category (see Section B.4).

The research community then took a radically different step. If network simulators are unable to interact with mobility simulators, they should be replaced by simplistic off-the-shelf discrete even simulators which could do this task. Accordingly, new simplistic network simulators were created, where the lack of elaborated protocol stacks was compensated by a native collaboration between the networking and the mobility worlds, as depicted in Fig. II-17.

MoVes [79] is an embedded system generating vehicular mobility traces and also containing a basic network simulator. The major asset of this project is its ability to partition the geographical area into clusters and parallelize and distribute the processing of the tasks from them, which improves the simulation performance. Although the mobility model reaches a sufficient level of detail, the project's drawback is the poor network simulation, which only includes a basic physical and MAC layer architecture and totally lacks routing protocols. In [80], authors also proposed an integrated vehic-

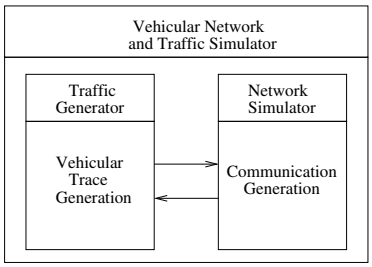


Fig. II-17. Interaction between Network and Traffic Simulators: The Integrated Case

ular and network simulator. As all solutions proposed by this approach, the authors developed their own traffic and network simulator. The vehicular traffic simulator is a synthetic model integrating basic microscopic motions where drivers may be in one of the following four modes: *free driving*, *approaching*, *following*, *braking*. A basic macroscopic model handles multi-lane and intersection management. Although being basic, this traffic model brings a sufficient level of details. However, the network simulator part is by far the major limitation of this project, as it is only modeled by a simplistic discrete event simulator handling a basic radio propagation and CSMA/CA MAC layer protocol.

As mentioned before, the major limitation of the embedded approach is actually the poor quality of the network simulator. Indeed, this approach has been so far only used to study basic network effects, but could not pass the test of recent mobile ad hoc routing protocols, including realistic and standardized physical and MAC layers. And this may also be intriguing, as the actual direction in network simulations is a specific compliance with standard protocols and computational efficiency through parallel and distributed computing.

Another approach recently carried on is to federate existing network simulators and mobility models through a set of interfaces (see Fig. II-18). For instance, MOVE [81] contains a single graphical user interface for the configuration the mobility modeling and network simulation. However, MOVE does not itself include a network simulator, but simply parses realistic mobility traces extracted from a micro-motion model SUMO [82] into a network simulator-dependant input trace format, then generates the appropriate scripts to be loaded by the network simulator. No interaction is therefore possible between the network simulator and the mobility model.

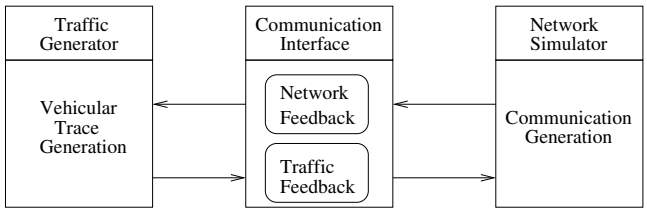


Fig. II-18. Interaction between Network and Traffic Simulators: The Federated Case

A different approach, taken by Prof. Fujimoto and his group in Georgia Tech [83] is

to generate a simulation infrastructure composed of two independent commercial simulation packages running in a distributed fashion over multiple networked computers. They federated a validated traffic simulator, CORSIM, with a state-of-the-art network simulator, QualNet, using a distributed simulation software package called the Federated Simulations Development Kit (FDK) [84] that provides services to exchange data and synchronize computations. In order to allow direct interaction between the two simulators, a common message format has been defined between CORSIM and QualNet for vehicle status and position information. During initialization, the transportation road network topology is transmitted to QualNet. Once the distributed simulation begins, vehicle position updates are sent to QualNet and are mapped to mobile nodes in the wireless simulation. Accordingly, unlike MOVE, both simulators work in parallel and thus may dynamically interact on each other by altering for example mobility patterns based on network flows, and vice and versa. The only limitation comes from the complex calibration of CORSIM and its large number of configuration parameters which must be tweaked in order to fit with the modeled urban area.

A similar solution has been taken by a team from UC Davis [85]. They developed a simulation tool federating the network simulator Swans and a synthetic traffic model. The complex vehicular flows are based on the Nagel and Schreckenberg model, extended to include lane changing in highway scenarios. The network simulator and the traffic simulator interact with each other by means of specific input and output messages.

Authors in [86] proposed *AutoMesh*, a realistic simulation framework for VANET. It is composed of a set of modules controlling all parts of a realistic simulation. It includes a *Driving Simulator Module*, a *Radio Propagation Module*, and *a Network Simulator Module*, all interlinked with feedbacks in order that any alteration made in one module influences the other modules. At the stage of the development of AutoMesh, the Driving Simulator Module only include random macro-movement and the IDM model for micro-movements. It is therefore unable to reproduce the non-uniform distribution of positions and speed usually experienced in urban area. However, the radio propagation module is very detailed, using 3D maps and digital elevation models in order to obtain a realistic radio propagation model in urban area.

Another promising approach is called *TraNS* [87] and also aims at federating a traffic simulator SUMO and a network simulator ns2. Using an interface called *Interpreter*, traces extracted from SUMO are transmitted to ns-2, and conversely, instructions from ns-2 are sent to SUMO for traffic tuning. TraNS will be extended to handle other network simulators such as Swans or Nab in the future. A similar project called MSIE [88] has been developed but using VISSIM instead of SUMO. This project is also more complete, as it proposes to interlink different simulators for traffic, network and application simulations. The major actual limitation is the communication latency between the different simulators and the expensive price of VISSIM. Besides, the interlinking interface itself is also not freely available at this time. Authors in [89] chose to replace VISSIM by a complete tool developed by themselves, the *CARISMA* traffic simulator. Although not being as complete as VISSIM or SUMO, it allows to accurately evaluate the effects of car-to-car messaging systems in the presence of urban impediments by benefiting from the federated approach and a "real-time" trip (re-)configuration.

By federating independent and validated simulators, the interlinking approach is able to benefit from the best of both worlds, as state-of-the-art mobility models may be adapted

to work with modern network simulators. However, it is computationally demanding, as both simulators need to be run simultaneously, and the development of the interface may not be an easy task depending on the targeted network and traffic simulators. Nevertheless, this is probably in this direction that most of the future pioneer work will come in the field of vehicular mobility modeling and networking.

The networking and mobility modeling community has a mutual interest in interacting between each others. Indeed, at the time of the promising benefits obtained from the various cross-layer approaches in network research, the ability to proactively or reactively act on mobility patterns in order to improve network efficiency or radio propagation, or even more promising, the ability to alter mobility patterns based on dynamically events radio transmitted will probably be a central approach in future networking research projects.

B.4 A Taxonomy of existing Synthetic VANETs Mobility Models

In this Section, we provide a taxonomy of existing VANETs synthetic mobility models and simulators freely available to the research community. We first introduce a set of criteria that will be able to better differentiate and classify the different synthetic models. We then provide a short summary of each model, including its assets and drawbacks, and provide its taxonomy in Table II-1 and Table II-2 according to the classification criteria. We purposely chose not include commercial-based traffic simulators as they cannot be freely used by the researcher working in the VANET field. As a consequence, most of the federated models described in the previous section may not be included. Similarly, we can neither add *Trace-based models* not *Survey-based Models* to our taxonomy as they are extrapolated from real mobility and cannot be classified according to our criteria.

Taxonomy Criteria

Prior to providing a classification, one need to define the criteria based on which to generate the taxonomy. The proposed criteria fall in three categories: **Macro-mobility**, **Micro-mobility**, and **Simulator Related**.

Macro-mobility Criteria

When considering macro-mobility, we do not only take into account the road topology, but also include trip and path generation, or even the effects of points of interests, which all influence vehicles movement patterns on the road topology. We therefore define the following criteria:

- *Graph* – The macro-motion is restricted to movements on a graph.
- *Initial and Destination Position* – The positions may be either random, randomly restricted on a graph or based on a set of attraction or repulsion points.
- *Trip Generation* – A trip may be randomly generated between the initial and destination points, or set according to an activity sequence.

- *Path Computation* – Provides the algorithms used to generate the path between the source-destination points contained in the trip.
- *Velocity* – The simulated velocity may be uniform, smooth or road-dependant.

Graphs

The selection of the road topology is a key factor for obtaining realistic results when simulating vehicular movements. Indeed, the length of the streets, the frequency of intersections, or the density of buildings can greatly affect important mobility metrics such as the minimum, maximum and average speed of cars, or their density over the simulated map. We categorize the graphs by the following criteria:

- *User-defined* – The road topology is specified by listing the vertices of the graph and their interconnecting edges.
- *Random* – A random graph is generated, which is often *Manhattan-grid*, *Spider*, or *Voronoi* graphs.
- *Maps* – The road topology is extracted from real maps obtained from different topological standards, such as *GDF*, *TIGER*, or *Arcview*.
- *Multi-lane* – The topology includes multi-lanes, potentially allowing lane changes, or not.

We show examples of the possible topologies in Fig. II-19.

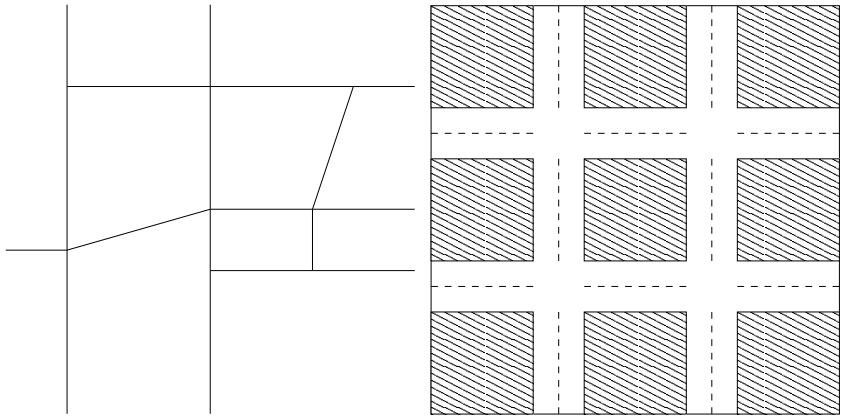
Attraction Points

Attraction or Repulsion points are particular source or destination points that have a potentially attractive or repulsive feature. For instance, for a weekly morning, residential areas are repulsion points and office buildings are attraction points, as a large majority of vehicles are moving from the former and to the latter. We depict the use of attraction points on a user-defined graph in Fig. II-20, where circles are entry/exit points of high-speed roads (thick lines), and squares are entry/exit points of normal-speed roads (thin lines).

Activity-based Trips

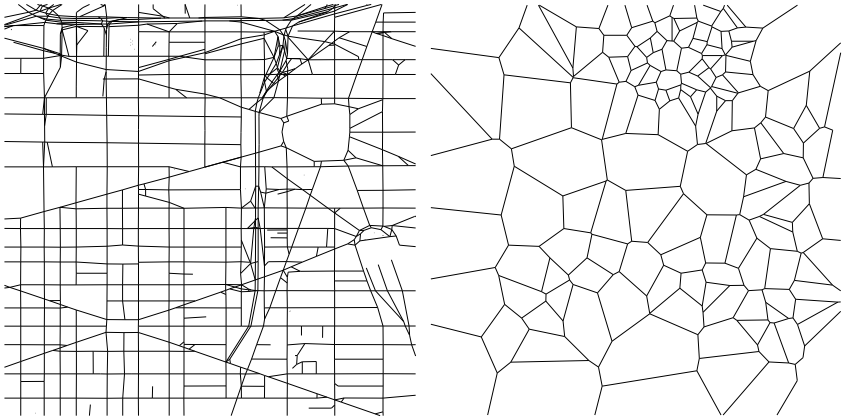
Activity sequences generation is a further restriction in vehicles spatial and temporal distributions. A set of start and stop points are explicitly provided in the road topology description, and cars are forced to move among them. In particular, multiple sets of points of interest can be specified, along with the probability matrix of a vehicle switching from one set to another. Fig. II-21 illustrates an activity sequence generated from a first order Markov chain between two categories of attractions points.

Micro-mobility Criteria In the proposed taxonomy, the micro-mobility aspect uses the following criteria:



(a) User-defined topology

(b) Manhattan-grid map topology



(c) TIGER map topology

(d) Clustered Voronoi

Fig. II-19. Road topologies examples

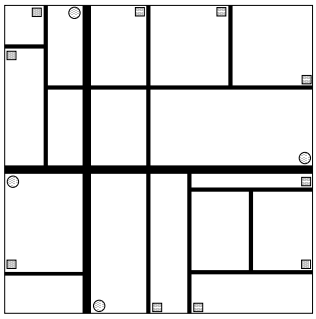


Fig. II-20. Example of Attraction Points on a User-defined graph

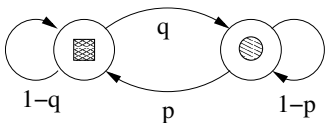


Fig. II-21. Activity-based Sequence between the attraction points in Fig. II-20

- *Human Mobility Patterns* – The car's internal motion and its interactions with other cars may be inspired from human motions described by mathematical models such as *Car Following*, or not.
- *Lane Changing* – Describes the kind of overtaking model implemented by the model, if any.
- *Intersections* – Describes the kind of intersection management implemented by the model, if any.

In this section, we shortly describe the most widely used vehicular specific micro-mobility models. We refer to [56] for a larger coverage of the different microscopic mobility models.

Car Following Models

The car following models are a class of microscopic models that adapts a following car's mobility according to a set of rules in order to avoid contact with the lead vehicle. A general schema is illustrated in Fig. II-22. Brackstone in [90] classified Car

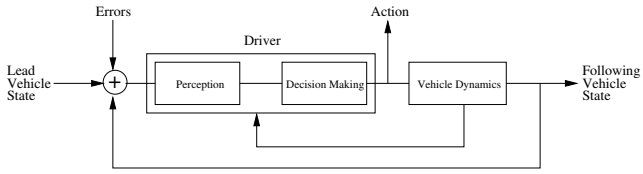


Fig. II-22. General Schema for Car Following Models

Following Models in five classes: *GHR Models*, *Psycho-Physical Models*, *Linear Models*, *Cellular Automata*, *Fuzzy Logic Models*. A description of the differences between those models is out of scope of this paper. We refer the interested reader to [91]. We only list here the widely used models in traffic simulations.

- Krauss Model (KM) [92]
- Nagel and Schreckenberg Model (N-SHR) [93]
- Wiedeman Psycho-Physical Model (Psycho) [94]
- General Motors Model (GM) [95]
- Gipps Model (GP) [96]
- Intelligent Driver Model (IDM) [97]

Lane Changing Models

Despite the large attention given to the driving tasks in general (such as Car Following Models), much less attention has been directed to lane changing. Modeling lane changing behaviors is a more complex task. Indeed, it actually includes three parts: the need

of lane changing, the possibility of lane changing, and the trajectory for lane changing. Each part is important to generate realistic lane changing models. And unlike car following models, it also needs to consider nearby cars and traffic flow information. Most of the models are based on a *Gap Acceptance* threshold [98] or a set of rules [99]. But recent approaches ([100, 101]) also considered forced merging, behavior aspects or game theory. Lane changing is not widely considered in open vehicular mobility models. In this survey, we mostly find

- Gibbs Model for Lane Changing (GP-LC) [98] and its variations
- Wiedeman Psycho-Physical Model for Lange Changing (Psycho-LC) [102]
- MOBIL [101]

Intersection Management

Intersection management adds handling capabilities to the behavior of vehicles approaching a crossing. In most cases, two different intersection scenarios are considered: a crossroad regulated by stop signs, or a road junction ruled by traffic lights. Nevertheless, all intersection management technics only act on the first vehicle on each road, as the car following model automatically adapts the behavior of cars following the leading one. The most basic ones consider intersections as obstacles and abruptly stop, yet more complex ones, such as the IDM_IM and IDM_LC [103], smoothly stop cars at stop-based crossing, or acquire the state of the semaphore in a traffic light controlled intersection. If the color is green, passage is granted and the car maintains its current speed through the intersection. If the color is red, crossing is denied and the car is forced to decelerate and stop at the road junction. Fig. II-23 illustrates the IDM_IM behavior when approaching an intersection with respect to the deceleration and the multi-lane management.

(a) Acceleration management

(b) Multi-lane management

Fig. II-23. Intersection management in IDM_IM and IDM_LC

Simulator Related Criteria Finally, we provide these additional criteria, which are more specific to the mobility simulator or to the interaction with a network simulator:

- *Obstacles* – The model considers radio obstacles, either in the form of an obstacle topology for network simulators and a propagation computation interface for

network simulators, or directly a radio propagation trace file.

- *Visualization* – The model includes a visualization tool.
- *Output* – Describes the kind of output generated by the mobility model, such as NS-2 or QualNet compliant traces.
- *Language* – Provides the programming language on which the simulator has been developed.

Taxonomy of Synthetic Vehicular Models

In this section, we simultaneously provide a brief description of the major synthetic mobility models available to the vehicular networking community, and classify them in Table II-1 and Table II-2 according to the previously defined criteria. As previously mentioned, we cannot include the *Trace-based* nor the *Survey-based* models as they have been obtained from real mobility and do not fall in the taxonomy. We include some *Traffic Simulator-based* models if they are based on freely available traffic simulators.

First, we point out that many realistic traffic simulation tools, such as PARAMICS [74], CORSIM [75], VISSIM [76] or TRANSIMS [77] have been developed to analyze vehicular mobility at both microscopic and macroscopic level with a very high degree of details. However, all the aforementioned softwares are distributed under commercial licenses, a major impediment to adoption by the academic research community. With the exception of few teams that developed parsers (e.g. [122, 123]), or federated a realistic traffic simulation tool with a network simulator (such as FDK [84]), these tools have been originally designed for traffic analysis and not for generation of movement traces usable by networking simulators. Furthermore, the presence of copyrights impedes the modification/extension of the source code when particular conditions, not planned by the original software, have to be simulated. For such reasons, we will not consider these tools in the following, their scope being very different from VANET mobility simulators are intended for. For a complete review and comparison of such traffic simulation tools, the interested reader can refer to [124].

When mobility was first taken into account in simulation of wireless networks, several models to generate nodes mobility patterns were proposed. The Random Waypoint model, the Random Walk model, the Reference Point Group (or Platoon) model, the Node Following model, the Gauss-Markov model, just to cite the most known ones, all involved generation of random linear speed-constant movements within the topology boundaries. Further works added pause times, reflection on boundaries, acceleration and deceleration of nodes. Simplicity of use conferred success to the Random Waypoint model in particular, however, the intrinsic nature of such mobility models may produce unrealistic movement patterns when compared to some real world behaviors. Despites, random models are still widely used in the study of Mobile Ad-hoc Networks (MANETs).

As far as Vehicular Ad-hoc Networks (VANETs) are concerned, it soon became clear that using any of the aforementioned models would produce completely useless results. Consequently, the research community started seeking more realistic models. The simple Freeway model and Manhattan (or Grid) model were the initial steps, then more

	Macro-Mobility							
	Graph				Init/Dest Position	Trip	Path	Velocity
	User Defined	Random	Map	Multi-lane				
Virtual Track [104]	yes		TIGER [105]	no	random	random S-D	RWP	uniform
IMPORTANT [106]		Grid		no	random	random S-D	RWM, RWalk	smooth
BonnMotion [107]		Grid		no	random	random S-D	RWM	uniform
RiceM [108]			TIGER	no	random	random S-D	Dijkstra	uniform
SUMO [82] MOVE [81] TraNS [87]	yes	grid, spider	TIGER	yes	random, AP	random S-D activity	RWalk, Dijkstra	smooth, road-dep
CARISMA [89]			ESRI [109]	yes	random	random S-D	Dijkstra, Speed, Density	smooth, road-dep
SHIFT [110]	yes			yes	AP	activity		smooth, road-dep
STRAW [111]			TIGER	no	random	random S-D	RWalk, Dijkstra	smooth
GrooveSim [112]			TIGER	no	random	random S-D	RWalk, Dijkstra	uniform, road-dep
Obstacle [113]		Voronoi		no	random	random S-D	Dijkstra	uniform
Voronoi [114]		Voronoi		no	random	random S-D	RWalk	uniform
GEMM [115]		Grid		no	AP	random S-D	RWP	uniform
CanuMobiSim [116]	yes		GDF [117]	no	random, AP	random S-D activity	RWP, Density, Dijkstra	uniform
City [118]		Grid		no	random	random S-D	RWM	smooth
MobiREAL [119]	yes			no	random	random S-D	RWalk	uniform
SSM/TSM [120]		Grid	TIGER	no	random	random S-D	Dijkstra	uniform, road-dep
MoVES [79]			GPSTrack [121]	no	random		RWalk	uniform, road-dep
Gorgorin [80]			TIGER	yes	random		RWalk	smooth
AutoMesh [85]			TIGER	yes	random	random S-D	Dijkstra, Density, Speed	uniform, road-dep
VanetMobiSim [78]	yes	Voronoi	TIGER, GDF	yes	random, AP	random S-D activity	RWP, Density, Dijkstra, Speed	smooth, road-dep

S-D: Source-Destination; *AP*: Attraction Point; *road-dep*: Road dependent;

TABLE II-1. MACRO-MOBILITY FEATURES OF THE MAJOR VEHICULAR MOBILITY MODELS

	Micro-Mobility				Simulator Related			
	Human Patterns	Intersection	Lane Changing	Radio Obstacles	Visualization Tool	Output	Platform	Remarques
Virtual Track [104]	no	no	no	no	no	ns-2, glomoSim, QualNet	C++	
IMPORTANT[106]	CFM	no	no	no	no	ns-2	C++	unrealistic CFM
BonnMotion [107]	no	no	no	no	yes	ns-2, glomoSim, QualNet	Java	
RiceM [108]	no	no	no	no	no	ns-2, glomoSim, QualNet	C++	
SUMO [82] MOVE [81] TraNS [87]	CFM (Krauss)	stoch turns	no	no	yes	ns-2, glomoSim, QualNet	C++	*federated* traf/net simulator, *validated* micromodel
CARISMA [89]	CFM (Krauss)	stop signs	no	yes	yes	ns-2, glomoSim, QualNet	C++	*federated* traf/net simulator
SHIFT [110]	CFM	no	LC	no	yes	none	C++/ SHIFT	config. CFM/LC
STRAW [111]	CFM (Nagel Schreck)	traffic lights, signs	no	no	no	Swans	JiST-Swans	
GrooveSim [112]	no	no	no		yes	none	C++	
Obstacle [113]	no	no	no	yes	yes	ns-2, glomoSim, QualNet	C++	
VoronoiM [114]	no	no	no	no	no	ns-2	C++	
GEMM [115]	no	no	no	no	no	ns-2	Java	
CanuMobiSim [116]	IDM	no	no	yes	yes	ns-2, glomoSim, QualNet,	Java	
City [118]	IDM	stoch turns	no	no	yes	ns-2	C++	
MobiREAL [119]	CPE	no	no	yes	yes	GTNetS	C++	pedestrian mobility
SSM/TSM [120]	no	traffic lights, traffic signs	no	no	no	ns-2	C++	
MoVES [79]	CFM (Psycho)	random traffic lights, traffic signs	no	no	yes	none	C++	*integrated* traf/net simulator
Gorgorin [80]	CFM (Psycho)	traffic lights, traffic signs	CFM (Psycho-LC)	no	yes	none	C++	*integrated* traf/net simulator, *validated* micromodel
AutoMesh [85]	IDM	stop signs	no	yes	yes	ns-2, glomoSim, QualNet	C++	*federated* traf/net simulator
VanetMobiSim [78]	IDM, IDM_IM, IDM_LC	traffic signs, traffic lights	MOBIL	yes	yes	ns-2, qualNet, glomoSim	Java	*validated* macro/micro model

CFM: Car Following Model; *IDM*: Intelligent Driver Model *CPE*: Condition-Probability-Event;
IDM_IM: IDM with Intersection Management; *IDM_LC*: IDM with Lange Changes;

TABLE II-2. MICRO-MOBILITY FEATURES OF THE MAJOR VEHICULAR MOBILITY MODELS

complex projects were started involving the generation of mobility patterns based on real road maps or monitoring of real vehicular movements in cities. However, in most of these models, only the macro-mobility of nodes was considered. Although car-to-car interactions are a fundamental factor to take into account when dealing with vehicular mobility [125], little or no attention was paid to micro-mobility. More complete and detailed surveys of mobility models can be found in literature [126, 127, 128, 129].

Recently, new open-source tools became available for the generation of vehicular mobility patterns. Most of them are capable of producing traces for network simulators such as *ns-2* [130], *GloMoSim* [131], *QualNet* [132], or *OpNet* [133]. In the rest of this section, we review some of these tools, in order to understand their strengths and weaknesses.

The IMPORTANT tool [106], and the BonnMotion tool [107] implement most of the random mobility models presented in [126], including the Manhattan model. This model restricts nodes macro-mobility on a grid, while the micro-mobility contains a *Car Following Model*. The BonnMotion does not consider any micro-mobility. When related to our proposed framework, we can easily see that the structure of both tools is definitely too simple to represent realistic motions, as they only model basic motion constraints and hardly no micro-mobility.

The GEMM tool [115] is an extension to BonnMotion's and improves its traffic generator by introducing the concepts of human mobility dynamics, such as *Attraction Points (AP)*, *Activity*, or *Roles*. Attraction points reflect a destination interest to multiple people, such as grocery stores or restaurants. Activities are the process of moving to an attraction point and staying there, while roles characterize the mobility tendencies intrinsic to different classes of people. While the basic concept is interesting, its implementation in the tool is limited to a simple enhanced RWM between APs. It however represents an initial attempt to improve the realism of mobility models by considering human mobility dynamics.

The MONARCH project [108] proposed a tool to extract road topologies from real road maps obtained from the TIGER [105] database. The possibility of generating topologies from real maps is considered in the framework, however the complete lack of micro-mobility support makes it difficult to represent a complete mobility generator. Indeed, this mobility model is simply a Random Waypoint Model restricted on a graph extracted from real topological cities. Although it brings some spatial correlations, it absolutely lacks time, car-to-car, and car-to-infrastructures correlations. Besides, the authors showed that their model was having similar patterns than the RWM, a proof of the lack of realism.

The Obstacle Mobility Model [113] takes a different approach in the objective of obtaining a realistic urban network in presence of building constellations. Instead of extracting data from TIGER files, the simulator uses random building corners and Voronoi tessellations in order to define movement paths between buildings. It also includes a radio propagation model based on the constellation of obstacles. According to this model, movements are restricted to paths defined by the Voronoi graph.

The *Simulation of Urban MObility (SUMO)* [82] is an open source, highly portable, microscopic road traffic simulation package designed to handle large road networks. The car microscopic movement model in SUMO is a car following model and includes

a stochastic traffic assignment modeled by a probabilistic route choice according to driver models. SUMO contains parsers for various topologies, ranging from TIGER, Arcview, or even VISSIM. Routes assignments may also be imported from various sources. However, at that time, SUMO is not able to output traces straightforwardly usable by network simulators.

The Mobility Model Generator for Vehicular Networks (MOVE) was recently presented [81]. It is a simple parser for SUMO and enhances its complex configuration with a nice and efficient GUI. MOVE also contains a parser to generate traces usable by network simulators such as ns-2 or QualNet.

SUMO is also the root functionality of TraNS [87], a federated model including ns2. Using an interface called *Interpreter*, traces extracted from SUMO are transmitted to ns-2, and conversely, instructions from ns-2 are sent to SUMO for traffic tunning. Accordingly, interactions between the vehicular traffic and network may be implemented.

Another important microscopic mobility simulator is the SHIFT Traffic Simulator [110]. It has been developed by the PATH Project at the UC Berkley, and is now a well established micro-mobility simulator that generates the trajectories of vehicles driving according to validated models on realistic road networks. More specifically, SHIFT is a new programming language with simulation semantics and was used in *SmartAHS* as means of specification, simulation and evaluation framework for modeling, control and evaluation of Automated Highway Systems (AHS). The major limitation of this simulator is its limitation to the modeling of segments of highways and its lack of complete topology modeling.

The CARISMA traffic simulator [89] is a realistic simulator containing microscopic and macroscopic features. It implements the Krauss's car following model, adds a stop sign intersection management, imports real topological maps in ESRI standard [109]. It furthermore provides a real-time trip management, which is a very interesting feature for the evaluation of car-to-car messaging. This model has also been interlinked with ns-2 for realistic evaluations of vehicle-to-vehicle messaging systems. One major limitation comes from the ESRI shape files, which are not publicly available unless one buy some ESRI products. Moreover, lane changing models and complex intersection managements are not considered at that time.

The Street Random Waypoint (STRAW) tool [111] is a mobility simulator based on the freely available Scalable Wireless Ad Hoc Network Simulator (SWANS). Under the point of view of vehicular mobility, it provides urban topologies extractions from the TIGER database, as well as micro-mobility support. STRAW is also one of the few mobility tools to implement a complex intersection management using traffic lights and traffic signs. Thanks to this, vehicles are showing a more realistic behavior when reaching intersection. The concept behind STRAW is very similar to the framework described in section B.1, as it contains accurate mobility constraints as well as a realistic traffic generator engine. STRAW also includes several implementations of transport, routing and media access protocols, since they are not present in the original SWANS software. The main drawback of the tool is the very limited diffusion of the SWANS platform.

The GrooveSim tool [112] is a mobility and communication simulator, which again uses files from the TIGER database to generate realistic topologies. Being a self-

contained software, GrooveSim neither models vehicles micro-mobility, nor produces traces usable by network simulators. The interesting feature of this model is the non uniform distribution of vehicles speeds. Indeed, motion constraints such as speed limitations, often force vehicles to give up in their effort to reach the velocity initially set by the model. Although that is might look as a straightforward pattern, this type of motion constraints is, at this time, considered only by few simulators. GrooveSim includes four types of velocity models, where the most interesting is the road-based velocity when used in conjunction with a shortest trip path generation. The authors illustrated how vehicles were naturally choosing the roads with the highest speed limitations while on their journey. The main drawback of this tool is however its lack of a micro-mobility model as well as the lack of mobility traces for network simulators.

The CanuMobiSim tool [116] is a tool for the generation of movement traces in a variety of conditions. Extrapolation of real topologies from detailed Geographical Data Files (GDF) are possible, many different mobility models are implemented, a GUI is provided, and the tool can generate mobility traces for *ns-2* and *GloMoSim*. Unlike many other tools, the CanuMobiSim tool keeps micro-mobility in consideration, implementing several car-to-car interaction models such as the *Fluid Traffic Model*, which adjusts the speed given vehicles local density, or the *Intelligent Driver Model (IDM)*, which adapts the velocity depending on movements between neighboring vehicles. Also unlike other tools, CanuMobiSim includes a complex traffic generator that can either implements basic source-destination paths using Dijkstra-like shortest path algorithms, or similarly to the GEMM, it can model trips between *Attraction Points* depending on the class of users' specific motion patterns. This solution is actually the only fully implemented and available solution considering heterogeneous classes of user and destinations. In order to improve its modeling capability, CanuMobiSim has even been recently extended (see [134]) by the same authors and now includes radio propagation information for ns-2 and GloMoSim/QualNet.

In recent months, a couple of research teams proposed a new set of simulators that comes closer to the objective to accurately model vehicles' specific motions. The first one is the City Model [118]. It has been basically designed for routing protocols testing and no network simulator traces are provided. This model includes a basic micro-mobility model composed of the IDM and a simplistic crossing management. Crossings are modeled like obstacles, where a car needs to reduce its speed and stop before the crossing. Then, the vehicle changes its direction according to a particular probability. This simulator mostly lacks modularity mostly due to its unique grid-based macro-mobility constraints, to the restriction to stochastic turns, and to the lack macro-mobility patterns based on human mobility dynamics.

The second is the SSM/TSM model [120]. It represents actually two different mobility models, a *Stop Sign Model* and a *Traffic Sign Model*. The motion constraints part is dealt using a TIGER parser, while the traffic generator includes the *Car Following Model*. As GrooveSim, both SSM and TSM include a road-dependent velocity distribution. However, this model goes farer than GrooveSim, since it contains a basic traffic generator which makes its mobility traces more realistic than GrooveSim's. And similarly to STRAW, SSM/TSM has been specifically designed to model vehicles' motions at intersections. The authors managed to show how a basic intersection management such as a simple stop sign was able to produce a clustering effect at those intersection. In urban environment, this effect is better known under the name *Traffic Jam*, and is

hardly represented in most of the actual simulators. But similarly to the City Model, the SSM/TSM also lacks macro-mobility patterns based on human mobility dynamics.

The Voronoi Model [114] is an illustration of how Voronoi graphs proposed by some simulators could be refined and improved to generate smoother roads. Unlike other mobility models including Voronoi tessellations, this *Voronoi Model* does not model roads as graph edges, but as Voronoi channels. A Voronoi channel is a spatial area obtained after multiple application of a Voronoi Tessellation algorithm. It provides a global moving direction, while keeping some degree of liberty in the local direction patterns. Most of this model contributions are on the improvement of the motion constraints component as a promising random topology generator, while the traffic generator engine is a simple implementation of a Random Walk within each Voronoi channel. However, this model's absolute lack of micro-mobility considerations and macro-mobility patterns based on human mobility dynamics, makes it unrealistic for vehicular mobility modeling.

All models presented in this section so far claims to be able to model realistic vehicular motion patterns. However, with the exception of SHIFT, none of them formally validated their patterns agaist real vehicular traces, or validated traffic simulators. *VanetMobiSim* [78], on the other hand, is the only synthetic model so far, which motion patterns for urban and highways environments have been validated. Indeed, the authors compared the traces obtained from VanetMobiSim and from CORSIM on similar urban configurations. They managed to show that the spatial distributions, the speed distributions, and the traffic shock waves generated by both models were similar. As CORSIM has been formally validated against real urban traces, so are VanetMobiSim's.

VanetMobiSim models car-to-car and car-to-infrastructure interactions, allowing it to integrate stop signs, traffic lights, safe inter-distance management and behavior based macro-mobility including human mobility dynamics. It also includes various road topology definitions, ranging from realistic GDF [117] or TIGER [105], to user-defined or random topologies. It lets the user define the trip generation between random source-destination, to activity-based trips. Moreover, the path used on the defined trip is also configurable between a Dijkstra shortest-path, a road-speed shortest path and a density-based shortest path. It finally generates traces for various network simulators. VanetMobiSim is at that time one of the most realistic and configurable synthetic model for the generation of vehicular motion patterns.

A special attention should also be brough to a novel solution named *MobiREAL* [119]. Although it focuses on the modeling of pedestrian mobility, its strict compliance with the proposed framework and its novel approach of cognitive modeling makes it very promising for a future extension to vehicular mobility. The most interesting features is that MobiREAL enables to change a node or a class of nodes' mobility behavior depending on a given application context. At this time, only CanuMobiSim, VanetMobisim and MobiREAL are able to include this feature. This particular application context is modeled by a *Condition Probability Event (CPE)*, a probabilistic rule-based mobility model describing the behavior of mobile nodes, which is often used in cognitive modeling of human behavior. As most of recent mobility models, MobiREAL is able to include geographical informations. Moreover, it is also able to use this information to generate obstacles and more specifically it is able to model radio's interference and attenuations on the simulation field. With CanuMobiSim's extension and the Obstacle model, they are the only models that are able to both generate motion traces and

signal attenuation information. MobiREAL's major drawback at this time is the limited diffusion of Georgia Tech Network Simulator (GTNets) and the manual configuration of all necessary parameters, which requires a full recompilation of the simulator at each reconfiguration.

Recently, new approaches appeared in realistic scalable simulations of vehicular mobility. In [79], the authors created MoVes, a complex mobility generator on top of Artis [135], a scalable distributed simulation middleware. MoVes features cars following models, drivers' characterizations, intersection management and includes a parser module to include GPS maps using the GPS TrackMaker program [121]. However, MoVes does not include any lane changing model, and no realistic path generation is supported.

Gorgorin [80] also integrated a network and a mobility simulator. Although the idea looks promising, the major flow at this time is the relative simplicity of both simulators. Indeed, although the mobility model is able to import TIGER maps and includes a similar micro mobility model to VISSIM, it does not consider any macro-mobility aspect. Moreover, similarly to MoVes, the network simulator also suffers from its simplistic architecture and from its poor diffusion compared to QualNet, OpNet or Ns-2.

The UDel Models [61] are a set of mobility and radio propagation models generated for detailed large-scale urban mesh networks. The urban mobility part is significantly different from all the previous approaches, as detailed urban vehicular and pedestrian mobility is based on surveys. Indeed, urban planning and the US Department of Labor generated a large database of statistics on time use or human mobility preferences. The generated simulator also considers a detailed urban propagation model and includes an accurate map builder capable of parsing GIS dataset and adding realistic radio obstacles.

C DISCUSSION

Although the benefits from using Mobility Reactive approach are obvious, they mostly depends on a protocol's ability to predict the local topology. We described in Section A the different approaches to generate predictions in order to improve adequacy. However, realism is also important as the performance of any prediction technique is closely related to its ability to follow nodes' mobility patterns. The performance is therefore related to the mobility model used during the simulation.

For example, it is quite easy to predict the Random Waypoint Model, as a node samples its origin, destination and velocity and then move following a straight line toward the destination point at a fixed velocity. Accordingly, if the node transmits those information, any neighboring node will be able to build a first order kinetic model based on the fixed velocity and straight trajectory. The mobility model and the prediction model are therefore adequate and the prediction efficiency maximized.

However, in the more realistic mobility models we described in Section B, a perfect match between the mobility patterns and the prediction model is mostly impossible. The larger is the divergence, the lower will be the prediction efficiency. So, iterative

prediction models are developed in order to reduce the gap between the prediction model and the mobility patterns though learning processes.

Accordingly, results obtained from prediction-based algorithms should be put in perspective to the mobility models used. An algorithm might be very efficient with the Random Waypoint model, but loose most of its assets when tested against a realistic vehicular mobility model such as VanetMobiSim. The mobility prediction and modeling worlds are therefore closely interlinked and should be jointly studied.

CHAPTER III

Kinetic Graphs in MANETs

Contents

Abstract—*The concept of Kinetic Graphs is an interesting example of the application of mobility predictions to graph theory. Unlike the static counterpart, graphs are assumed to be continuously changing and edges are represented by time-varying weights. Each node follows a posted* trajectory*, which may change at any moment through a* trajectory update*. Kinetic graphs are a natural extension of classical ones and provide solutions to similar problems, such as convex hulls, spanning trees or connected dominating sets. In this chapter, we propose a natural representation of the trajectories, then describe the protocol, the data structure and the overhead involved by the* posting *of the trajectories. We also propose two time-varying weights which could be used to adapt graph theory algorithms to the kinetic aspect. Finally, we define methods to keep track of changes in the topology without requiring to periodic beaconing.*

Keywords—*Kinetic graph, kinetic data structure, trajectory, time varying weight, MANET*

KINETIC Graphs is particular class of graphs aimed at maintaining the attributes of interests, such as spanner, Voronoi tesselation, or convex hull, in graphs with moving vertices. In regular graphs subject to mobility, the position of the vertices are updated and then the attribute of interest is recomputed. Since the attributes do not changes homogeneously in the whole graph, it is hard to find a refreshing rate which optimizes the computing cost of the reconfiguration. Unlike this fixed step approach, kinetic graphs perform an event-driven simulation where only events relevant to the vertex and the attributes are generated and processed. Kinetic graphs are therefore mobility proactive, as the structure is updated only when an attribute that effectively alters the graph is changed.

Kinetic graphs are also a special application to a more generic approach called the *Kinetic Data Structures (KST)* introduced by Bash et al. [136]. In the KDS framework, it is assumed that trajectories of objects are known, but the algorithm does not know when trajectories will change. This is a direct use of mobility predictions applied to data structures, where two goals are targeted: the *optimality* with respect to the attribute, and the *maintenance efficiency*. This topic has been widely studied in various area such as mobile facility locations [137], clustering and routing [138] or shortest path [1]. A survey on KDS can be found in [139].

Mobile Ad Hoc Networks (MANETs) are an emergent concept in view of infrastructureless communication. It appeared clear to the community that graph theory heuristics such as Connected Dominating Sets (CDS), Convex Hulls, or Minimum Spanning Trees (MST), could be applied to various objectives such as Broadcasting, Routing, or Topology Control. Moreover, any algorithm should be distributed as no central omniscient entity exists in MANETs. However, due to the limited capability of processing power, storage and energy supply, many conventional algorithms are too complicated to be implemented in wireless ad hoc networks. Thus, they require efficient distributed algorithms with a low computation complexity and a low communication complexity. More importantly, distributed algorithms should also be *localized*, as each node running the algorithm could only use the information of nodes within a constant number of hops. However, localized algorithms are difficult to design or even sometime impossible. The community however started to work on adapting decades of graph theory outbreaks and to solve several challenging questions such as *localized Delaunay tri-*

angulation [140], *localized spanner [141]*, *localized spanning trees [142]* and even to broadcasting [143, 144]. A survey on Localized approaches for broadcasting and topology control may be found in [145], and for routing in [10].

When looking at the state of the art of achievements in the approaches described in the previous paragraphs, we can see a straightforward interweaving aspect. *Localized Protocols* solve the performance issues of *Kinetic Structures*, and conversely, *Kinetic Structures* provide solutions to handle mobility for *Localized Protocols* for MANETs. Unfortunately, these two communities have worked quite independently, and only few works [1, 138] appeared to have seen the potential benefits from joining both worlds.

A Summary of Contribution

In this Chapter, we are going to describe the *Kinetic Graphs* framework and illustrate how to adapt well known mobility protocols (topology control, broadcast, or routing) to the Kinetic approach. We will first provide a general description of how the trajectories are modeled, how the structure is initially built and finally, how it is maintained. We emphasis that our objective is to suppress the periodic beaconing process widely used by almost all mobility protocols in order to adapt to mobility. Then, we will discuss two different kinetic criteria that could be easily adapted in most of the protocols for MANETs. Finally, we will illustrate a successfully application of the Kinetic Graph approach to Topology Control.

B Organization of Work

The Chapter is organized as follows. In Section C, we propose a possible trajectory representation in Kinetic Graphs, while in Section D, we present the neighborhood discovery phase involving a common packet format and a data compression for geo-localization information. Section E describes two possible time varying link weights for the construction of Kinetic Graphs, and Section F provides heuristics in order to aperiodically maintain the neighborhood. Finally, Section G illustrates an application of Kinetic Graphs to Topology Control and Section H concludes the Chapter.

C Trajectory Knowledge

In order to model trajectories in Kinetic Graphs, we need to define kinematic 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 chapter, 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 as described in Chapter II. 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 [146] or [147] 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 GSP 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

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

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

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

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 (III-1), a node is able to compute the future position of its neighbors, and by using (III-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

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

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

[1] 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.

[3] Up to normalization, a Unit Disk Graph (UDG) corresponds to a graph where every two nodes are connected if and only if they are at a distance at most the homogenous transmission range.

D NEIGHBORHOOD DISCOVERY

Basically, the *Kinetic Graph* neighborhood discovery procedure makes a node detect changes in its neighborhood without any exchange of periodical beacon messages. During this phase, each node broadcasts a single[4] $Hello$ message indicating its presence in the neighborhood, and transmitting its mobility parameters x, y, dx, dy, along with its stability parameters β and t_0. 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.

D.1 Geo-localization Data Format

In 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.

GPS Data Format

According to the GPS standard, the 3D positioning provides the coordinates of a GPS device in a 3-axis referential, whose origin is the gravity center of the GPS satellite constellation. Then, the GPS terminal converts this raw data into exploitable *longitude*, *latitude*, and *elevation* in the World Geodetic System 84 (WGS84) [148], the most widely used providing a worldwide navigational system. The provided data format is as follows:

- **longitude–** describes the location of a place on Earth *east* or *west* of the Greenwich meridian. A longitude is expressed in sexagesimal notation as $23° \ 27' \ 30" \ E$. An alternate representation is a decimal representation of the minutes and degrees $23.45833°$, where the Est/West suffix is replaced by a negative sign for coordinates west of the Greenwich Meridian. Accordingly, a longitude may be represented by a signed floating point ranging in $[+180°,-180°]$ usually with a 6 digits precision.

- **latitude–** describes the location of a place on Earth *north* or *south* of the *Equator*. Similarly to the longitude, a latitude is expressed in sexagesimal notation as $13° \ 19' \ 43" \ N$, with an alternate decimal representation $13.32861°$, where the North/South suffix is replaced by a negative sign for coordinates south of the

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

Equator. Accordingly, a latitude may be represented by a signed floating point ranging in $[+90°, -90°]$ usually with a 6 digits precision.

- **elevation–** describes the altitude of a place on Earth relative to the WGS-84 ellipsoid. The elevation is therefore expressed in by a signed integer ranging from $8000m$ (Mount Everest) to $-11000m$ (Mariana Trench).

The common point of the tree coordinates is that they are usually represented by 32 bits. Accordingly, each geo-localization set is usually represented by a 96 bits or 12 bytes.

GPS Time Representation

In order to precisely determine the position of a GPS device, its internal clock must be synchronized with the satellites atomic clocks. The GPS system therefore provides a global synchronization mean to any application connected to a GPS device.

GPS time is expressed as a number of seconds since the beginning of the GPS epoch on Sunday January 6th 1980 at 0:00 UTC. Initially represented by a 32 bit integer, this value has been increased to a 64 bits long integer at the end of the last century. Accordingly, the transmission of the time in a packet requires 64 bits or 8 bytes.

D.2 A Common Geo-localization Message Format

This section defines the content and the structure of a mobility message containing a configurable set of geo-localization or mobility information.

All `<mobility>` messages are conformed to the following specification:

```
<mobility> = <value-semantic><value>
```

`<value-semantic>` is an 8 bit field that describes the structure of the `<mobility>` tag.

- *bit 0 (position bit):* Messages with this bit cleared ('0') do not contain the position of the node. Messages with this bit set ('1') contain position information.
- *bit 1 (velocity bit):* Messages with this bit cleared ('0') do not contain the velocity of the node. Messages with this bit set ('1') contain the velocity.
- *bit 2 (azimuth bit):* Messages with this bit cleared ('0') do not contain the azimuth of the node. Messages with this bit set ('1') contain the azimuth.
- *bit 3 (stability bit):* Messages with this bit cleared ('0') do not contain the stability of the node. Messages with this bit set ('1') contain the stability.
- *bits 4-7 are RESERVED*

<value> is a field containing the mobility parameters. The length of this field may be obtained from the <value-semantic> field.

<value> = <pos><azi><velo><stab><time>

where

- <pos> is a 48 bit field containing the coordinates of a node following the general layout <pos> = <Longitude><Latitude><Elevation>.
- <velo> is an 8 bit field containing the node's velocity in m/s
- <azi> is an 8 bit field containing the node's azimuth in degree
- <stab> is an 8 bit field containing the node's stability. It represents the node eagerness to keep the current mobility parameters.
- <time> is an 16 bit field containing the GPS time in seconds when the mobility parameters have been sampled.

The basic layout of a <mobility> message included in a HELLO packet is illustrated in Fig. III-1.

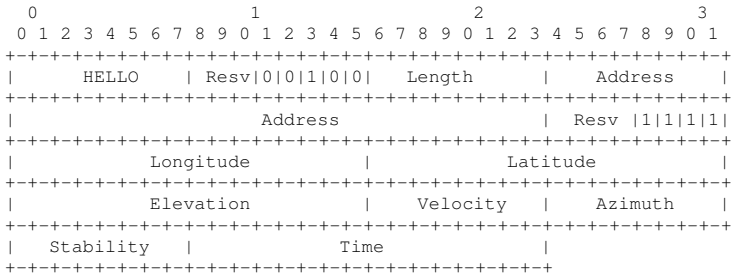


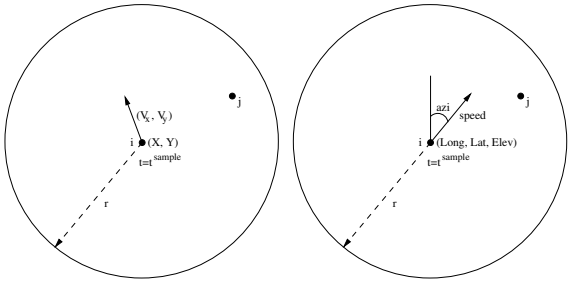
Fig. III-1. Hello Packet Containing Geo-localization Information

D.3 The Real Overhead of Diffusion of Geo-localization Data

In this section, we illustrate the overhead generated by the transmission of geo-localization data in wireless ad hoc network. We show the non negligeable increase of the size of mobility control packets compared to conventional ones.

We first describe a typical mobility information format. As depicted in Fig. III-2, node i transmits its position and velocity to node j. For that matter, it transmits 5 fields: X, Y, V_x, V_y and t^{sample} completely describing its localization, velocity, heading and freshness. In this case, those 5 fields are represented by integers and encoded into 32 bits. The total payload per mobility information is therefore 168 bits (21 bytes).

If the system uses GPS coordinates, the same message consists of the following 5 fields $longitude$, $latitude$, $speed$, $azimuth$, t^{sample}. In that case, the first four fields



(a) Cartesian (b) GPS

Fig. III-2. Neighborhood discovery typical message content

are represented by 32 bits integers or floating points, while the last field is represented by a 64 bit integer. The total payload per mobility information is therefore 200 bits (25 bytes).

Accordingly, considering a simple neighborhood discovery heuristic which involves a one-hop restricted broadcast of neighborhood information (such approach is used for example in the MPR protocol by OLSR [4]), the total control traffic depends on each node's neighbor degree. Fig. III-3 illustrates the drastic overhead increase for the transmission of a single packet as a function of the nodes density. We compare the two previous examples in contrast with the conventional single ID approach. We model a density ranging from 1 neighbor per node to 20 neighbors per node, which is a reasonable assumption for sparse and dense networks.

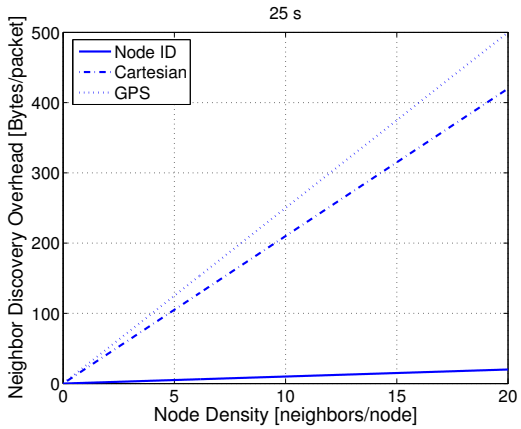


Fig. III-3. Illustration of the per packet neighbor discovery overhead with geo-localization data

We can clearly see the drastic increase in the overhead per packet as a function of the node density. A typical application usually generating periodic transmission of such packet, we can also extrapolate these results for the overhead created by the use of geo-localization information in mobile ad hoc networks.

D.4 Reducing the Geo-localization Overhead

The motivation of our approach comes from the observation that geo-localization and time data use non-appropriate representation formats. For example, mobility information, if even considered, are usually represented by Cartesian coordinates encoded by a 32 bit integer potentially covering a simulation area of 2^{32} square meters, which is clearly never reached in practice. The representation of longitude or latitude are also done by a 32 bit integer (see [105]), even though the maximum value may only be $180 \cdot 10^6$, including a 6 digit precision. Our approach therefore substitutes the standard representation with a more efficient one based on a *mantissa/exponent* number representation. We aim at using the minimum number of bits required to cover the full range of applicability of geo-localization data.

Compressing GPS Coordinates

As GPS coordinates are represented by a signed integer ranging up to $180 \cdot 10^6$, we use 16 bits. For practical reasons that we will explain later, we reserve 1 bit for the sign code c. Of these 15 remaining bits, the most significant 8 bits represent the mantissa a, and the least significant 7 bits represent the exponent b. In the following, K is a constant that is common to all nodes implementing this compression algorithm. As the geo-localization data is represented by an integer, we set $K^{geoloc} = 1$.

Algorithm 1 Signed Integer Compression

Require: A signed integer i and $K > 0$
Ensure: A compressed unsigned integer

```
a=b=0
c = SIGN(i) {Returns 1 if c < 0; Returns 0 if c ≥ 0}
j = ABS(i) {To use it as unsigned}
while j/K ≥ 2^b do
    b++
end while
b-
if b < 0 then
    a = 0
    b = 0
else if b > 127 then
    a = 255
    b = 127
else
    a = 256 · (j/(K·2^b) − 1)
end if
if c > 0 then
    return (a ·128 + b) | 0x8000
else
    return (a ·128 + b) & 0x7FFF
end if
```

Algorithm 2 Signed Integer Decompression

Require: Compressed unsigned integer i and $K > 0$.
Ensure: A signed integer

```
1: c = (i ≫ 15 ) & 0x01
2: j = i & 0x7FFF {To remove the sign bits}
3: a= j ≫7
4: b= j − (a ·128)
5: return (1 + a/256) · 2^b · K · (−1)^c
```

Using this method, the minimal value representable is $(-1)^c \cdot K$ and the maximum value is $(-1)^c \cdot 3.39 \cdot 10^{38} \cdot K$. We finally add the sign bit representation to obtain the 16 bit representation of a 32 bit signed integer.

This method may therefore be used to represent the geo-localization information in GPS or Cartesian coordinates with a 50% reduction of the number of bits without loss of precision.

Compressing GPS time

As GPS time is represented by an unsigned integer whose maximal value is 2^{64}, we also use 16 bits. The most significant 8 bits represent the mantissa a, and the least significant 8 bits represent the exponent b. Using this method, the minimal value representable is K and the maximum value is $1.15 \cdot 10^{77} \cdot K$. Similarly to the geo-localization case, as GPS time is represented by an integer, we set $K^{time} = 1$.

Algorithm 3 Unsigned Integer Compression

Require: An unsigned integer i and a constant $K > 0$
Ensure: A compressed unsigned integer

```
1: a=b=0
2: while i/K ≥ 2^b do
3:    b++
4: end while
5: b−
6: if b < 0 then
7:    a = 0
8:    b = 0
9: else if b > 255 then
10:   a = 255
11:   b = 255
12: else
13:   a = 256 · ( i/(K·2^b) − 1 )
14: end if
15: return (a · 255 + b)
```

This method may therefore be used to represent the GPS time representation using a 75% reduction of the number of bits without loss of precision.

Algorithm 4 Unsigned Integer Decompression

Require: Compressed unsigned integer i and a constant $K > 0$.
Ensure: An unsigned integer

1: $a = i \gg 8$
2: $b = i - (a \cdot 256)$
3: **return** $\left(1 + \frac{a}{256}\right) \cdot 2^b \cdot K$

The same approach may be used for the speed, azimuth or stability, representing a and b with 4 bits each. Using this method, the minimal value representable is 0 and the maximum value is $63488 \cdot K$. By fixing a specific K to reach a target precision, we are therefore able to reduce the size of the transmission of this data by 75%.

As the stability is represented by a strictly positive integer, we propose to fix $K^{stab} = 1$.

The speed representation should consider the application use. It is represented in meters per second. Therefore, by considering a 2 digits precision, we reach a range between $0m/s$ and $634m/s$, providing a sufficient range for its representation. Accordingly, we fix $C^{speed} = 0.01$.

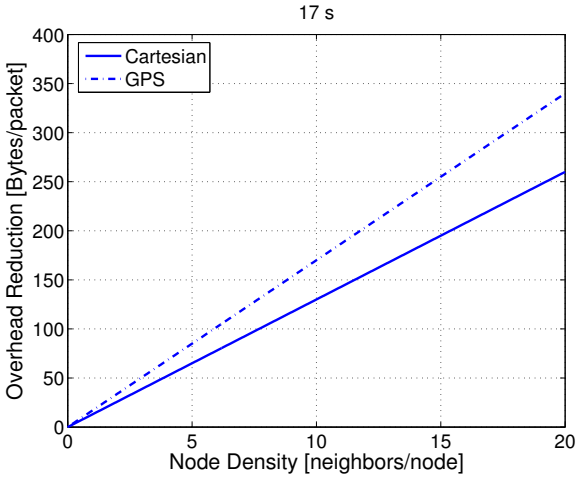
The azimuth needs further precisions. In literature, the azimuth is represented in degrees with a 6-digit precision, ranging between $0°$ to $360°$. As such time of format cannot be represented by the 8-bit compression, we could increase the size of a and b. However, similarly to the velocity, we should also analyze the applications using this information. The azimuth is directly used as projection in order to obtain the direction of movement. By using the 8 bit compression, we obtain a 2-digits precision, which generates a 0.015% error in cosine projection and 0.006% in sine projection. Accordingly, we are convinced that the error generated by the loss of 4 precision digits are negligeable and therefore assume a loss-less azimuth compression. We therefore set $K^{azimuth} = 0.01$

Scenarios	Size			Value	Compr.
	a [bit]	b [bit]	c [bit]	K	
GPS	7	7	1	1	50%
Cartesian	7	7	1	1	50%
Time	8	8	0	1	75%
Azimuth	4	4	0	0.01	75%
Speed	4	4	0	0.01	75%
Stability	4	4	0	1	75%

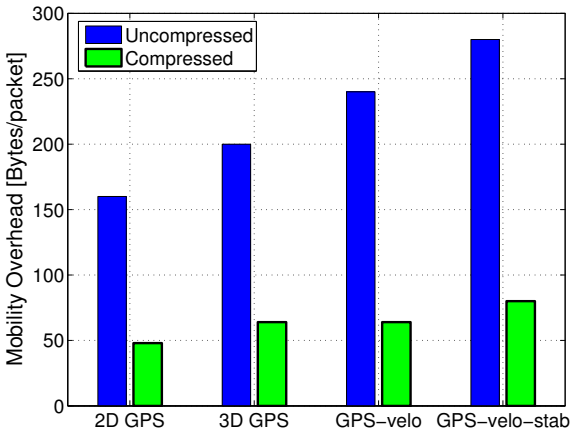
TABLE III-1. SUMMARY OF THE PARAMETERS FOR DIFFERENT MOBILITY DATA

Figure III-4 illustrates the compression efficiency. We first show in Fig. III-4(a) the overhead reduction as a function of the node density for a 2D geo-localization including velocity and could see the non negligeable benefit achieved by our proposed solution. Second, we describe in Fig. III-4(b) the compression efficiency depending on the format of the geo-localization data transmitted for an average density of $10nbrs/node$. We can also clearly see that already reaching a 66% reduction for a simple 2D geo-localization, the efficiency reaches 71% when we transmit 2D geo-localization, veloc-

ity, azimuth, time and stability.



(a) As a function of node density



(b) Depending on the data format

Fig. III-4. Compression Efficiency

Finally, in Fig. III-5, we illustrated the Routing Overhead Ratio of the OLSR [4] routing protocol as a function of the node density. We used CBR traffic at a rate of 50kb/s with 10 sources and control traffic of 0.5 hello_pkt/s. We can clearly see that as the density increases, so does the cost of carrying geo-localization data. Yet, the proposed compression is able to significantly reduce this drawback.

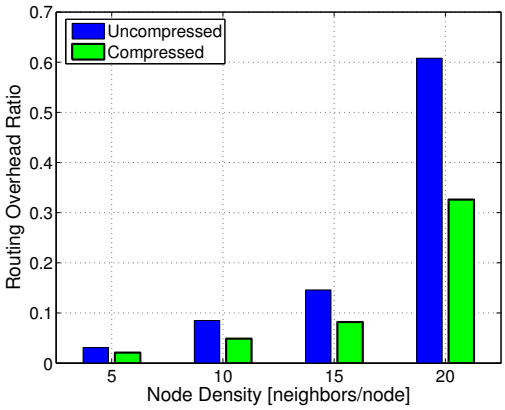


Fig. III-5. Illustration of the routing overhead ratio of OLSR with geo-localization information

D.5 Discussion

In this section, we discussed the overhead generated by the transmission of geographic localization information for neighborhood discovery. We proposed a compression algorithm managing to reduce this overhead up to 71%. We also introduced a Wireless Vehicular Communication message format defining a structure for the transmission of mobility information. We finally illustrated how our solution could improve the routing overhead of the OLSR protocol up to 46%. We therefore provided a framework for optimized and configurable transmission of geo-localization information and believe our approach could ease interoperability and improve the performance of location-based solutions in ITS. Our solution has also been proposed for a possible standardization within the IETF [149].

E 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 time varying link 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 chapter, it is important that graph protocols be distributed and local. Accordingly, we suggest as potential targets localized graph constructions described in [145].

E.1 Kinetic Distance Weight

The *power cost* function, required to transmit between nodes i and j at time t, is defined as $P_{ij}(t) = D_{ij}^{\alpha}(t) + \gamma$, where $\alpha \geq 2$ and for some constants γ. 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

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

as the power cost function for the weight of the *Kinetic Graphs*. By choosing the distance between nodes as the link weight, one obtains minimum power routes that help preserve battery life (see Figure III-6).

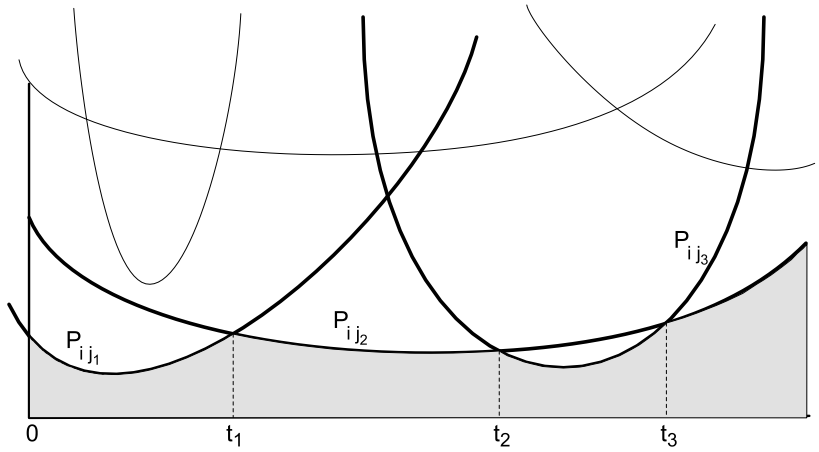


Fig. III-6. The power function, where each parabola represents the energy needed to reach each neighbor of node i

We then define

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

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 III-7).

Assuming independent node trajectories,

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

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

[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

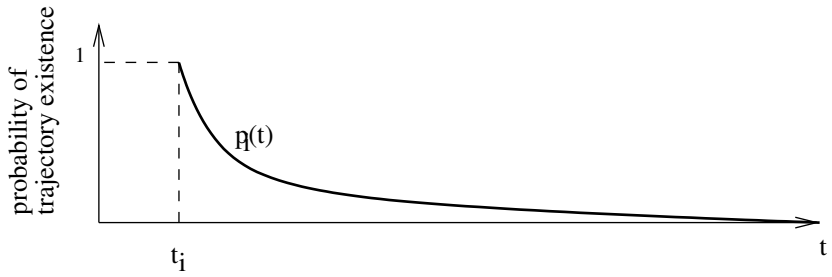


Fig. III-7. The stability function, where the probability for a node i to behave as predicted decreases exponentially

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

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

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 C.

We finally define

$$W_{ij}(t) = -\frac{p_{ij}(t)}{P_{ij}(t)} \cdot Sigm_{ij}(t) \tag{III-8}$$

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

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

as the composite link weight between two neighbors (see Figure III-8). 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 Fig. III-9-C. Node i tries to find the best next hop node to reach a far destination node. To do

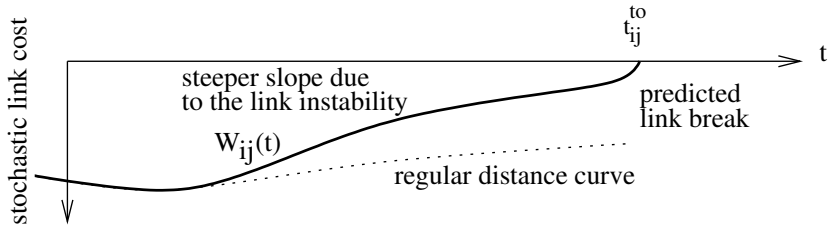


Fig. III-8. The composite link weight function, where we can see the cost increase due to the link's instability

so, 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. Fig. III-9-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 Fig. III-9-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.

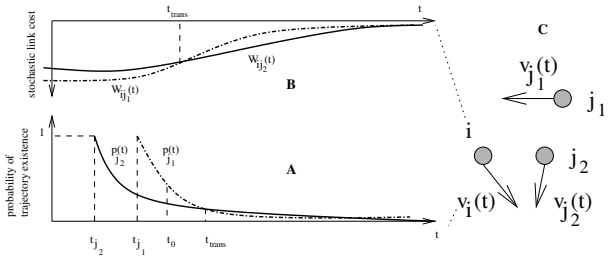


Fig. III-9. Topology example

E.2 Kinetic Nodal Degree Weight

In Graph theory, besides the Euclidian distance, the nodal degree is also widely used, as it provides a high 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 bigger diffusion potential.

Similar to the euclidian distance, the nodal degree may also be applied to Kinetic Graphs as a time varying link weight. We explain in the next, the method for modeling *Kinetic Nodal Degrees* in MANETs. Similarly to Section C, we model nodes' positions as a piece-wise linear trajectory and, as we show in Chapter IV, the corre-

sponding trajectory durations are lengthy enough to become a valuable cost for using kinetic degrees.

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

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

Consequently, thanks to (III-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, solving

$$\begin{aligned} D_{ij}^2(t) - r^2 &= 0 \\ a_{ij}t^2 + b_{ij}t + c_{ij} - r^2 &= 0, \end{aligned} \tag{III-11}$$

gives 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 Fig. III-10).

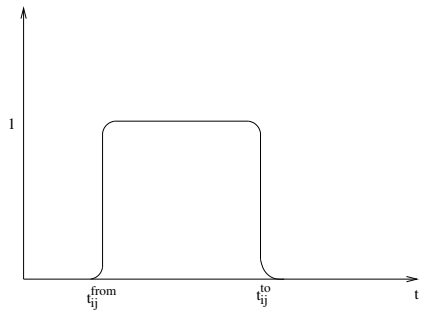


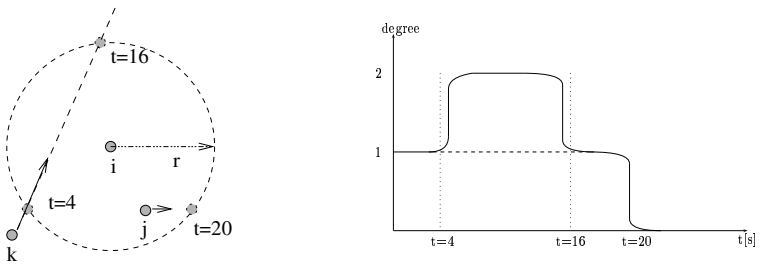
Fig. III-10. Double sigmoid function modeling a link lifetime between node i and node j

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

$$Deg_i(t) = \sum_{k=0}^{nbrs_i(t)} \left(\frac{1}{1+\exp(-a\cdot(t-t_k^{from}))} \cdot \frac{1}{1+\exp(a\cdot(t-t_k^{to}))} \right) \tag{III-12}$$

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 (III-12), each node is able to predict its actual and future degree and thus is able to proactively adapt its coverage capacity. Figure III-11(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, Fig. III-11(b) illustrates the evolution of the kinetic degree function over t.



(a) Node i kinetic neighborhood

(b) Node i kinetic nodal degree

Fig. III-11. Illustration of nodes kinetic degrees

Finally, the kinetic degree is obtained by integrating (III-12)

$$\widehat{Deg}_i(t) = \int_t^{\infty} \left(\sum_{k=0}^{k=nbrs_i} \left(\frac{1}{1+\exp(-a\cdot(t-t_k^{from}))} \cdot \frac{1}{1+\exp(a\cdot(t-t_k^{to}))} \right) \right) \tag{III-13}$$

For example, in Fig. III-11(b), node i kinetic degree is ≈ 32.

Similarly to the previous section E.1, the kinetic nodal degree may also be stochastically weighted by the probability of the existence of the link. The last task is therefore to consider the uncertainty of a predicted degree by adding the stability function (III-6). Accordingly, we obtain a criterion reflecting nodes actual and future degree, yet biased by the uncertainty of the link between all of their respective neighbors.

By using substituting (III-6) to (III-13), we define

$$\widehat{Deg}_i(t) = \int_t^{\infty} \left(\sum_{k=0}^{k=nbrs_i} \left(\frac{1}{1+\exp(-a\cdot(t-t_j^{from}))} \cdot \frac{1}{1+\exp(a\cdot(t-t_j^{to}))} \cdot \exp(-(\beta_i+\beta_j)(t-\frac{t_i\beta_i+t_j\beta_j}{\beta_i+\beta_j})) \right) \right) \tag{III-14}$$

Using the same topology as Fig. III-11 and applying the uncertainty of predicted degrees, we obtain a stochastically predicted nodal degree depicted in Fig. III-12. 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 Fig. III-12 left 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 Fig. III-11. Worse, its decreasing

curve is sharper than the one between nodes i and j taken alone. Similarly to Fig. III-11, 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.

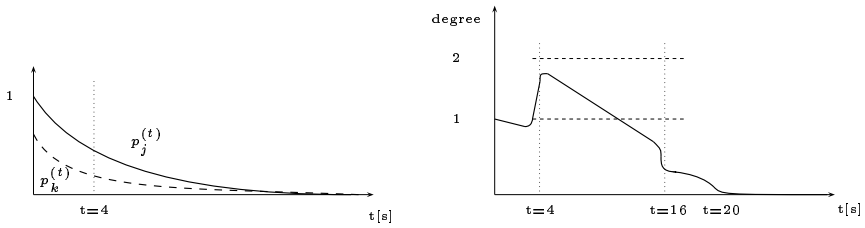


Fig. III-12. Stochastically Predicted Nodal Degree

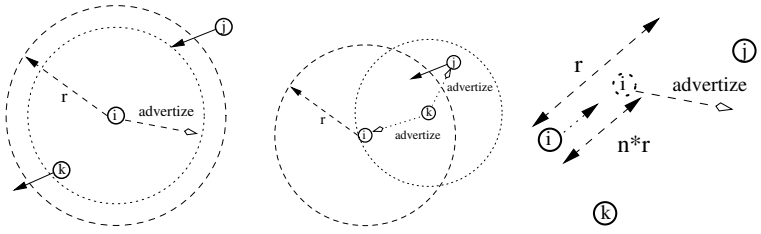
F ADAPTIVE APERIODIC NEIGHBORHOOD MAINTENANCE

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 multipoint relays.

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 III-13(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 III-13(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 III-13(c));

All three heuristics may be implemented simultaneously, further improving the capability to detect nodes stealthily entering other nodes neighborhood. The adaptive coverage



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

Fig. III-13. Three heuristics to detect incoming neighbors in a per-event basis

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 Thesis, we chose an adjusting factor of $n = \frac{2}{3}$ and postponed the tweaking of the adjusting factor to future work.

A second approach is identical to the information exchange period proposed in [142]. 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 [142] 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 not a single 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

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

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

G KINETIC TOPOLOGY CONTROL IN MANETS

While most topology control protocols only address limited network mobility, we propose in this Section a quasi-localized topology control algorithm, denoted as Kinetic Adaptive Dynamic topology control for Energy efficient Routing (**KADER**), that considers mobility predictions and Kinetic Graphs in order to construct and maintain a power efficient topology without relying on periodic beacons.

G.1 Background on Topology Control in MANETs

A mobile ad-hoc network (Manet) consists of a collection of mobile nodes forming a dynamic autonomous network through a fully mobile infrastructure. Nodes communicate with each other without the intervention of centralized access points or base stations. In such a network, each node acts as a host and may act as a router. Due to limited transmission range of wireless network interfaces, multiple *hops* may be needed to exchange data between nodes in the network, which is why the literature often uses the term of *multi-hop* network in Manet. The *topology* of a multi-hop network is the set of communication links between nodes used by routing mechanisms. Removing redundant and unnecessary topology information is usually called *topology control*.

The importance of topology control lies in the fact that it critically affects the system performance in several ways. For once, as shown in [150], it affects networks spatial reuse. Choosing a power assignment too large results in excessive interference while choosing it too small creates a disconnected network. Power assignment in topology control also exerts influence on the energy consumption of communication, thus impacts battery life which is a critical resource in many mobile applications. In addition, topology control also improves contentions at the MAC layer.

Several topology control algorithms have been proposed [151, 152, 153, 154, 155, 156, 157] to create power efficient network topologies in MANETs, yet only considering limited mobility or no mobility at all. Some of the algorithms require explicit propagation channel (e.g., [154]), while others (e.g., [151]) incur significant message exchanges. In [154] and its extension [158], Rodoplu *et al.* introduced the notion of *relay region* and *enclosure* for the purpose of power control. It is shown that the network is strongly connected if each node maintains links with the nodes in its enclosure and the resulting topology is a minimum power topology. In [153], Ramanathan *et al.* presented two centralized algorithms that minimize the maximum power used per node while maintaining the (bi)connectivity of the network. In the same paper, the authors also proposed two distributed heuristics for mobile networks. LINT uses locally available neighbor information collected by routing protocols to keep the degree of

neighbors bound. LILT further improves LINT by overriding the threshold on the node degree when topology changes indicated by routing updates result in undesirable connectivity. In [151], Narayanaswamy *et al.* developed a power control protocol, called COMPOW, that reduces the power level to a common value that reaches a maximum network connectivity. In [152], authors proposed an algorithm, called $CBTC(\alpha)$, in which each node finds the minimum power p such that transmitting with p ensures that it can reach some nodes in every cone of degree α. Other works also exist on power efficient topology control. Following a probabilistic approach, Santi *et al.* derived the suitable common transmission range which preserves network connectivity [159]. In [160], a "backbone protocol" is proposed to manage large wireless ad hoc networks, in which a small subset of nodes is selected to construct the backbone. In [142], Li *et al.* presented a MST-based topology control algorithm, in which each node builds its local minimum spanning tree (LMST) independently and only keeps on-tree nodes that are one-hop away as its neighbors in the final topology. This approach has been improved in [155] and in [156], where a MST is quasi-locally built from the LMST structure. Finally, [157] presents a strictly local protocol, called XTC, which does not only work on Unit Disk Graphs, but also on general weighted networks graphs.

G.2 Basic Idea

In this section, we are focusing on a novel approach, called stochastic mobility prediction, where nodes are able to predict their neighbors' future positions. We are adapting this concept to a topology control protocol denoted as Kinetic Adaptive Dynamic topology control for Energy efficient Routing (**KADER**). We base our approach on the DDR [161] protocol which we modified to obtain a distributed self-maintained topology control strategy which tries to optimize the power assignment for multi-hop transmissions, and where modifications to the topology are announced by the respective nodes in a per-event basis. The contribution of this Chapter includes: (i) KADER builds a self adaptive forest which maintains the network connectivity; (ii) each tree in the forest forms a zone, in which shortest path routes are proactively maintained; (iii) the criterion to build the forest is based on the relative power needed to reach a neighbor, thus minimizing the power assignment and creating a backbone adapted to mobility; and (iv) Since KADER is based on mobility predictions, it only needs to update its structure when a node changes its trajectory. Since most of links remain valid after a localized topology changes, the updates are also kept local further minimizing the maintenance cost. The capability of forming a self-adaptive topology that is closely linked to nodes relative mobility is what make KADER achieve *linear time and message complexity, scalability and energy efficiency*.

Both CONNECT and its extension are centralized algorithms that requires global information, thus cannot be directly deployed in the case of mobility. On the other hand, LINT and LILT cannot guarantee the preservation of the network connectivity. In opposite, KADER does not require global information and is able to ensure network connectivity. Moreover, COMPOW is known to give poor performance in the case of uneven spatial distributions, while the performance of KADER is not subject to the spatial distribution, and as a matter of fact, is especially well-suited in the case of uneven spatial distributions. Finally, what makes KADER unique compared to previous approaches, is its ability to use mobility predictions to maintain its backbone in a com-

plete per-event way (i.e. non periodically). Indeed, most of the proposed protocols either do not consider mobility induced topology changes, or perform this task periodically although trying to adapt the frequency of topology updates to nodes limited mobility ([142]).

G.3 KADER's Topology Construction Algorithm

We propose to construct a self-adapting forest from an ordinary network that consists of non-overlapping dynamic trees, thereafter called zones[7]. Each zone is kept connected with its neighboring zones through gateway nodes, thus making the whole network a set of connected zones. The size of a zone will increase or decrease dynamically without any need of periodic maintenance. Unexpected topological changes are announced by the respective nodes through a specific non-periodic message communicating its new mobility parameters. Following this event, the forest will adapt itself to the new topology.

The algorithm described hereafter consists of five cyclic time-ordered phases: *neighborhood discovery, preferred neighbor election, forest construction, self-adaptive intra-zone clustering,* and *self-adaptive inter-zone clustering.* Since these phases are similar to DDR [161], we will only describe the major extensions between DDR and KADER.

Neighborhood Discovery

Basically, KADER's 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 $Hello$ message indicating its presence in the neighborhood, and transmitting its mobility parameters x, y, dx, dy, along with its stability parameters β and t_0. Such message is emitted using maximum 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.

Preferred Neighbor Election

A node's *Preferred Neighbor (PN)* is a dedicated neighbor through which a node sends, receives, or forwards packets. It therefore represents the link on which a node sends its traffic. The criterion determining this neighbor depends on the application needs. For example, it could be the nodal degree in order to improve broadcast, or nodes energy level and traffic level for load balancing. It also could be a combination of all three. In KADER, we wish to obtain a criterion that is able to satisfy two objectives. The first one is to represent the energy needed to reach a neighbor. The second one

[7] We will later use the term tree and zone interchangeably.

is neighbor stability. A node's stability is the probability that it evolves as predicted. Since KADER does not periodically update its set of links, we want the links chosen by KADER to remain as stable as possible such that routing errors could be kept low. Accordingly, KADER should not elect the closest neighbor, but might decide to choose a more distant yet a more stable one. Therefore, KADER is able to lower nodes power assignment by preferring close-by neighbors, which increases transmission concurrency and improves nodes lifespan. It is also able to lower topology maintenance and routing errors by favoring stable nodes.

As described in Section E.1, We define

$$W_{ij}(t) = - \frac{p_{ij}(t)}{P_{ij}(t)} \tag{III-15}$$

$$= - \frac{e^{-(\beta_i+\beta_j)(t-\frac{t_i\beta_i+t_j\beta_j}{\beta_i+\beta_j})}}{a_{ij}t^2+b_{ij}t+c_{ij}} \tag{III-16}$$

as the composite link weight between two neighbors.

Election Algorithm

Based on the criterion described above and similarly to [161], a node i can determine its PN j at time t_1, which represents the time at which j has the smallest cost function over all k other neighbors of i.

$$PN_i(t_1) = j \quad iff \quad W_{ij}(t_1) = \min_{k \in nb_i} (W_{ik}(t_1))$$

Yet, since we are performing mobility predictions, a PN is not elected for a single time instant t, but for a time interval $[t_1, t_2]$. During this interval, also called *activation*, the link function assigned to this PN is the smallest over all k other neighbors. An activation between node i and node j over an interval $[t_1, t_2]$ is defined as

$$act(i,j)[t_1,t_2] = \begin{cases} \min t_1 \ s.th. \ PN_i(t_1) = j \\ \max t_2 \in (t_1;\infty) \ s.th \ PN_i(t) = j \end{cases}$$

Therefore, the set (i, j,$act(i,j)[t_1,t_2]$) uniquely identifies a *preferred link* between node i and node j activated from t_1 to t_2, and will thereafter mentioned as $\overline{ij}_{[t_1,t_2]}$.

Then, by always considering the smallest link weight function for all time t, i creates a set of actual and future preferred neighbors that always minimize the link weight.

Forest Construction

For every time instant, a dynamic forest, or a group of non overlapping dynamic trees, is constructed by connecting each node to its PN (similarly formulated as connecting the set of all preferred links). Due to mobility issues, PNs may change (along with preferred links) during the simulation. But since every node knows in advance the set of its actual and future PNs, thus actual and future preferred links, no future reconfiguration or exchange of messages is needed to adapt the topology. We prove in

the appendix that, whatever the network topology is, this approach always yields to a forest at each time instant.

In order to construct *preferred links* and consequently the forest, each node generates a table called *Intra-Zone table* (see Table III-2). Indeed, as soon as node i determines the set of its PNs, it must notify its neighbors, especially its PNs, of its decision. It first updates its *Intra-Zone table* by adding the set of its PNs in a *PN* field along with their respective activations. Any node appearing in the *PN* field of node i's *Intra-Zone* table means it is either a PN of node i, or it elected node i as PN. Node i then sends a *PN* message $PN_i = (i, \overline{ij}_{[t_1,t_2]})$. This message indicates that node i is electing node j as its PN with the activation $act(i,j)[t_1,t_2]$. Upon reception of i's message, node j checks whether it has been chosen as the PN of i. If so, it also updates its intra-zone table regarding i. Since nodes i and j appear in the *PN* field of their respective *Intra-Zone* table, a tree branch is built between node i and its preferred neighbor j during $[t_1, t_2]$. Therefore, those edges become a preferred link, and the set of preferred links in each neighborhood generates the set of preferred paths in the network.

We illustrated KADER's constructed dynamic forest in Figure III-14, considered at time $t = 0$. Full lines represent actual preferred links, while dashed lines future ones. For example, as we can see on the same Figure and on Table III-2(a), node k has a PN c activated between $t = 0s$ and $t = 10s$. Since the simulation time $t = 0$, the preferred link $\overline{kc}_{[0,10]}$ is active. But, node k also has a future PN d activated between $t = 10s$ and $t = 20s$. Therefore, at time $t = 0$, the preferred link $\overline{kd}_{[10,20]}$ is not yet activated and is considered as a future preferred link and depicted as a dashed line. It will be quietly activated at time $t = 10$.

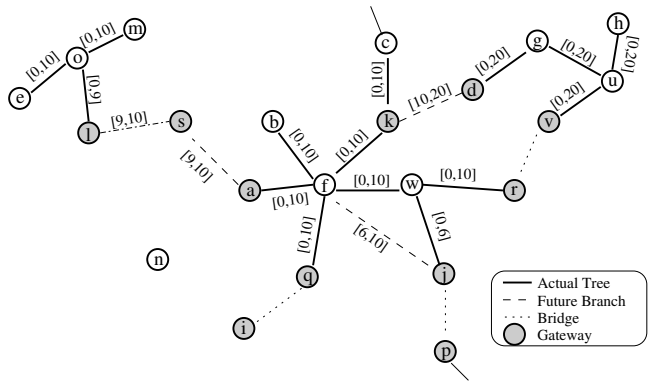


Fig. III-14. Constructed forest by KADER, considered at time $t = 0$. The brackets represent the links' activation intervals.

Self-Adaptive Intra-Zone Clustering

KADER aims at regrouping closed-by nodes into a zone in order to provide energy efficient communications. At this phase, we illustrate how nodes obtain the best path, with respect to the electing criterion, to reach all nodes in their zone and proactively maintain them in their intra_zone table. The process is very similar to [161], and readers may refer to this paper for a more detailed description.

When a node i gets elected by a neighbor j, it then locally notifies all its neighbors of this election. To do so, i sends a so called *Learned_PN* message $Learned\_PN_i = (i, \overline{ij}_{[t_1,t_2]})$, indicating that node j with $act(i,j)[t_1,t_2]$ has node i as its PN. Upon reception of this message, each tree member updates its intra_zone table, and re-advertises to its neighbors if it is not a leaf node[8]. For this purpose, each node generates another field in its intra_zone table called *Learned Preferred Neighbor* (*Learned_PN*, see Table III-2) in order to keep nodes that have been learned to belong to the same tree. Therefore, if node i is chosen to be the PN of j over a time interval $[t_1, t_2]$, j sends a *PN* message to inform its neighborhood of its elected PN. Among the neighboring nodes of j, the PN i forwards j's decision to each node that holds a tree edge with i, say node k, activated over a time interval $[t_3, t_4]$. Then, the local view of k's tree is that, over the time interval $([t_1, t_2] \bigcap [t_3, t_4])$, j is reachable through i. For example, in Figure III-14 and in Table III-2(b), node j elected node w as PN for a time interval $[0, 6]$. Node j is then a *Learned_PN* of node f over the time interval $([0, 6] \bigcap [0, 10]) = [0, 6]$.

Consequently, as we can see in Table III-2, on convergence of KADER, the *PN* field represents the next hop nodes to reach any node belonging to its zone (appearing in the *Learned_PN* field). This is a very interesting feature for routing since the end-to-end delay for route discovery may be limited.

PN	Learned_PN
$\mathbf{f}_{[0,10]}$	$a_{[0,10]}, b_{[0,10]}$
	$q_{[0,10]}, w_{[0,10]}$
	$r_{[0,10]}, j_{[0,10]}$
	$l_{[9,10]}, s_{[9,10]}$
$\mathbf{d}_{[10,20]}$	$g_{[10,20]}, u_{[10,20]}$
	$v_{[10,20]}, h_{[10,20]}$
$c_{[0,10]}$	-

(a) $Intra\_ZT_k$

PN	Learned_PN
$w_{[0,10]}$	$r_{[0,10]}$
	$j_{[0,6]}$
$\mathbf{k}_{[0,10]}$	$c_{[0,10]}, d_{[0,10]}$
$j_{[6,10]}$	-
$b_{[0,10]}$	-
$a_{[0,10]}$	$s_{[9,10]}, l_{[9,10]}$
$q_{[0,10]}$	-

(b) $Intra\_ZT_f$

TABLE III-2. INTRA-ZONE TABLE OF NODES k AND f REGARDING FIGURE III-14

Self-Adaptive Inter-Zone Clustering

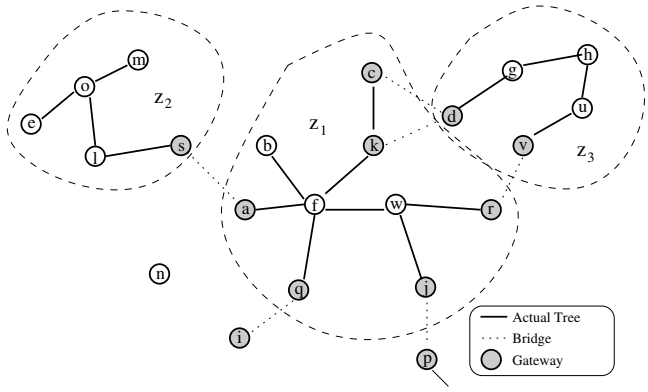
Once zones have been built, KADER's task is to keep them connected to each others at each time instant. For that matter, KADER uses a different table, called *Inter_Zone* table, that regroups connections to different surrounding zones. Each node belonging to this table is referred as *gateway* and the link connecting two zones as *bridge* (see Figure III-15). Moreover, each gateway must keep a connection to each of its peer-gateways belonging to its neighboring zones. We prove in the appendix that the zones created by KADER are always connected to each others if the original graph is complete.

At the beginning, neighbors of a node i are put in its *Inter_Zone* table during their full initial activation, say $[T_1, T_2]$, which is defined as the connection lifetime between the two nodes or the time two nodes remain direct neighbors. Then, as node i succeeds to add a neighbor j to its tree and updates its intra_zone table over an activation $[t_1, t_2]$,

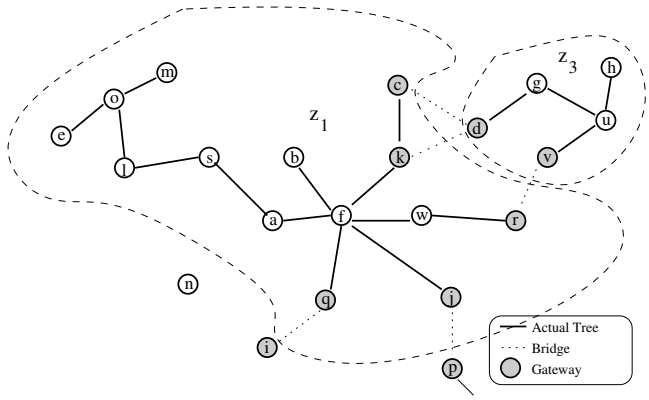
[8] A leaf node is a node which only has a single neighbor and which is never a PN.

it prunes j's initial activation. The remaining activation is then $\{[T_1, T_2] \backslash [t_1, t_2]\}$. During this time, node j is still not considered part of i's tree. Node j then appears in the *Intra_Zone* table over $[t_1, t_2]$ and in the *Inter_Zone* table over $\{[T_1, T_2] \backslash [t_1, t_2]\}$.

For example, in Figure III-15 and corresponding tables in III-3, node s is in the *Inter_Zone* table and also in the *Intra_Zone* table of node a, yet for different time intervals. For the time interval $[0, 9]$, node a is a gateway node and the link $\overline{sa}$ is a bridge. However, from the time interval $[9, 10]$, zone $z1$ grows and zone $z2$ fusions with it. Accordingly, for this time interval, node s becomes a *PN* and nodes l, o, m, e *Learned_PN* for node a. As we can see, the size of different zones grows and shrinks over time, which makes them self-adapting to the mobility-based topology changes, yet without exchanges of messages.



(a) Constructed Forest from t=0s to t=9s



(b) Construes Forest from t=9s to t=10s

Fig. III-15. KADER Constructed Forest and its evolution over time

Gate. ID	Zone ID
$s_{[0,9]}$	$z2$

(a) Inter_zone table of node a regarding Figures III-15(a) and III-15(b) .

PN	Learned_PN
$\mathbf{f}_{[0,10]}$	$c_{[0,10]}, b_{[0,10]}$
	$q_{[0,10]}, w_{[0,10]}$
	$r_{[0,10]}, j_{[0,10]}$
	$k_{[0,10]}$
$\mathbf{s}_{[9,10]}$	$l_{[9,10]}$

(b) Intra_zone table of node a regarding Figures III-15(a) and III-15(b).

TABLE III-3. INTER-ZONE AND INTRA-ZONE TABLES OF NODES a REGARDING FIGURE III-15

Quasi-local topology maintenance

In this subsection, we show how a particular message, called *New Trajectory* (NT), is used to inform neighboring nodes of any topology changes, and to trigger a quasi-local maintenance process. Therefore, KADER's zone maintenance can be seen as per-event based.

As mentioned before, node trajectories information remains valid during a short period of time. Then, since a node is unable to predict the time its neighbors will change their trajectories, it biases the link weight W to reflect the decreasing probability of the link existence, and to ensure that a good low power but unstable link could not be chosen for routing. Yet, we still consider this link valid as long as it is not otherwise notified. Consequently, when a node is changing its trajectory, it must inform its neighbors about the induced topology change. To do so, it sends a *New Trajectory* (NT) message to all its neighbors and piggybacks its new coordinates and velocity. Therefore, its neighbors are able to adapt their trees to this event. Eventually, the algorithm carries out the PN election phase again.

G.4 Analysis of KADER's Topology

In this section, we prove that under the assumption of a Unit Disk Graph (UDG) and a fully connected initial topology, the backbone created by KADER always leads to a tree and the trees are interconnected to form a forest.

Definition 6: An arbitrary undirected time dependent graph $G(t)$ is defined as $G(t) = (V, E(t))$, where V is the set of vertices, and $E(t)$ is the set of edges at time t.

Theorem 1: For any graph $G(t)$, let $G'(t) = (V, E'(t))$ be the subgraph obtained by connecting each vertex V to its preferred links $E'(t)$. Then $G'(t)$ is a forest.

Proof: Let $G(t)$ be the original graph at time t, and let $G'(t)$ be the graph obtained by executing the KADER algorithm for each vertex $v \in V$ at time t. We first recall that the main idea is to select for each node $v \in G(t)$, a neighbor that has the maximum link function *W*. In order to prove that $G'(t)$ does not contain any cycle

$C = v_i, v_{i+1}, \ldots, v_{i-1}, v_i$, let us suppose the contrary, and let v_i be the vertex of C with the biggest W.

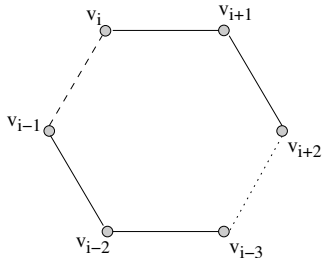


Fig. III-16. The proof of theorem 1

Let us consider two vertices of v_{i-1} and v_{i+1} adjacent to v_i in C (Figure III-16). Without loss of generality, assume that the algorithm on v_i chosen an adjacent vertex v_{i+1} (if neither v_{i-1} nor v_{i+1} had been chosen, C is not a cycle). Consider now the execution of the algorithm on v_{i-1}. We will show that such node will not choose v_i, thus implying that C is not a cycle.

Lets define $f(v_i, v_{i+1})$ as the weight function W between v_i and v_{i+1}. Since v_i chooses v_{i+1}, $f(v_i, v_{i+1}) > f(v_i, v_{i-1})$. And by v_{i-2}'s decision to choose v_{i-1}, $f(v_{i-2}, v_{i-1}) > f(v_{i-2}, v_{i-3}) > f(v_i, v_{i+1})$, f being a monotone increasing function. Therefore, $f(v_i, v_{i-1}) < f(v_{i-1}, v_{i-2})$. This proves that v_{i-1} will not choose v_i as PN, and C will not be a cycle. ■

Theorem 2: For any PN activation $act(v_i, v_{i+1})[t_1, t_2]$, and any graph $G'(t_1, t_2) = (V, E'(t_1, t_2))$ obtained by connecting each vertex to its preferred links $E'(t_1, t_2)$ activated during $[t_1, t_2]$, $G'(t_1, t_2)$ is then always a forest at every time instant included in $[t_1, t_2]$.

Proof: When a node v_i elects a PN v_{i+1} during an activation $act(v_i, v_{i+1})[t_1, t_2]$, it means that $\forall t \in [t_1, t_2]$, $f(v_i, v_{i+1})(t) > f(v_i, v_k)(t)$, $\forall k$. Since nodes share a common clock, all their current left activation are equal to the current time and will thereafter be considered as 0, past activations being irrelevant.

If a node v_i elects a PN v_{i+1} during an activation $act(v_i, v_{i+1})$ $[0, t_1]$, without loss of generality, v_{i+1} can elect a node v_{i+2} as PN during an activation $act(v_{i+1}, v_{i+2})[0, t_2]$. Since the algorithm prunes the activation between v_i and v_{i+2} as $([0, t_1] \bigcap [0, t_2])$, we must consider two cases. In the first case, the initial activation $act(v_i, v_{i+1})$ $[0, t_1]$ is less or equal than $act(v_{i+1}, v_{i+2})[0, t_2]$, thus it is kept unaltered during the forwarding steps. In the second case, the algorithm prunes the initial activation. The forwarded activations are two separated and mutually exclusive activations.

Let us consider $t_2 \leq t_1$. Then, following the development in the proof of Theorem 1, at some point, node v_{i-1} could elect node v_i during an activation $act(v_i, v_{i-1})[0, t_3]$, as $t_3 \leq t_1$. $[0, t_3]$ is the remaining activation after multiple pruning at each node in the path. Then, it means that $\forall t \in [0, t_3]$, v_{i-1} could elect v_i as PN, thus creating a cycle during this time. Theorem 1 prove that this situation is not possible, since $\forall t \in [0, t_3]$, we obtain a stable tree which is not a function of t. Then, during the activation $act(v_i, v_{i-1})[0, t_3]$, C does not contain any cycle.

Since the initial activation has been pruned, we still need to consider the case of the remaining activation ($[t_3, t_1]$). Without loss of generality, let us consider that this activation has been pruned at a single node v_{i+2}. This node has the possibility to elect v_{i+1} as PN (mutual election), updating the mutual activation as the union of their respective ones. Note that this case does not create a cycle. v_{i+2} can otherwise elect another node, say v_{i+3}. Since $[t_3, t_1] \bigcap [0, t_3] = \emptyset$, $\overline{v_{i+2}v_{i+3}}$ is then a branch of a different and independent tree and the situation is independent from the previous one. Therefore, this neither creates a cycle, which concludes the proof. ■

Theorem 3: ∀ G, let G' be the subgraph obtained by connecting each node to its preferred links during their respective activations. Then G' is a forest at every time instant.

Proof: ∀ node v_i, since all its PNs activation intervals are mutually exclusive ($\bigcap (act(v, v_1), \ldots, \ act(v, v_n)) = \emptyset$), from Theorem 2, we can conclude that KADER always yield to a forest at every time instant. ■

Theorem 4: For any complete graph G(t), the subgraph $G'(t) = (V, E'(t))$ created by connecting each vertex to its preferred neighbor creates a set of connected zones.

Proof: Let us consider the contrary and take two unconnected zones A and B. At any time instant, a neighbor of a node either belongs to its *Intra_Zone* table or to its *Inter_Zone* table. The former means both nodes belong to the same zone, and the latter means both nodes belong to two adjacent zones which they are connecting. Therefore, if a link between two nodes does not exist between two zones in $G'(t)$, it also could not exist in the original graph $G(t)$. This is a contradiction to the hypothesis of a complete graph $G(t)$. ■

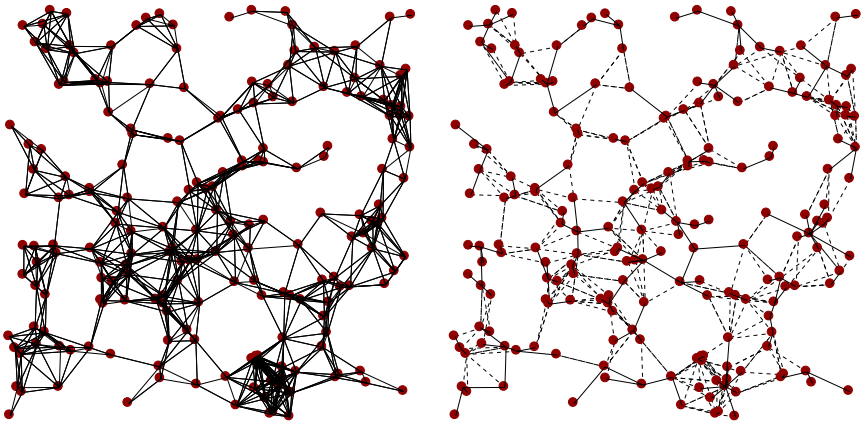
G.5 Properties of KADER's Topology

In this section, we study KADER's computed backbone with parameters such as average zone diameter (i.e. in term of number of hops), average number of zones in the network, average ratio of remaining edges, average ratio of PNs in the network, and average power assignment. The following results are obtained by measuring the metrics after the population of mobile nodes was distributed uniformly on a $A \times A$ grid where $A = 2000m$, with each node having a transmission range r of 250m. Moreover, each node has a different stability value, but nodes' average stability is $1/\beta = 30s$. We will compare KADER in two different cases : *variable density* and *globally constant density*[9].

We begin by illustrating in Figure III-17 the topology created by KADER from an arbitrary graph G (see Figure III-17(a)) to a forest and trees (see Figure III-17(b)), where solid lines are tree edges and dashed lines are bridges connecting different trees.

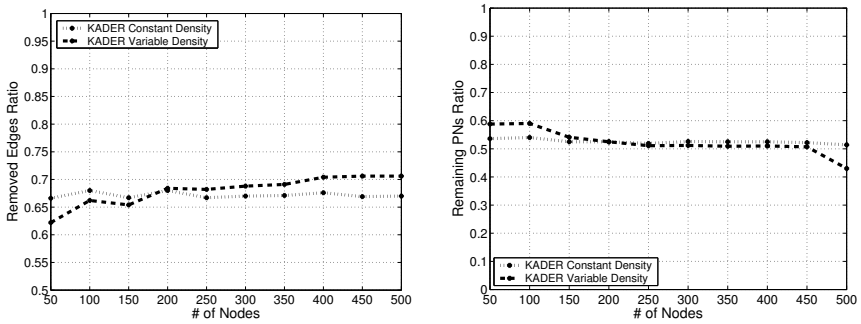
We can see in Figure III-18(a) that KADER is able to remove 65% of the total number of edges. By getting rid of unnecessary links, KADER improves spatial reuse and concurrent communications. Then, in Figure III-18(b), KADER is able to remove 45% of PNs, which helps to reduce the broadcasting overhead in the network. Therefore, by

[9]Globally constant density is obtained by maintaining the ratio $\frac{\#nodes \cdot \pi \cdot r^2}{A^2}$ fixed

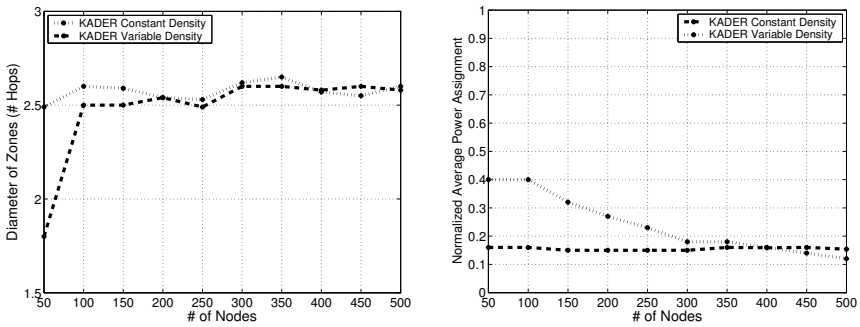


(a) An arbitrary graph (b) Constructed forest

Fig. III-17. KADER's Topology



(a) Removed Edges Ratio (b) Remaining PNs Ratio



(c) Diameter of Zones (d) Average Power Assignment

Fig. III-18. Properties of KADER's Topology

removing 65% of unecessary links and by only keeping a backbone of 55% of router nodes yet keeping the graph connected, KADER helps performing load reduction and proves to be broadcast efficient.

Then, the graph in Figure III-18(c) shows the diameter of a zone versus the number of nodes in the network. The diameter of a zone is defined as the length of the path that has the longest euclidean distance. If the zone diameter is obtained, then we can place an upper bound on the Intra_zone end-to-end delay. We clearly see in Figure III-18(c) that zones in KADER are relatively stable, in both a variable and a constant density. This comes from its distance parameter in the electing criterion. Since KADER always tries to create a link between nearby neighbors, neither density nor the number of nodes have a big impact on the zone's diameter. This shows another interesting property of KADER that is its suitability for dense and sparse networks on non-evenly distributed nodes.

Finally, Figure III-18(d) illustrates KADER's average node power assignment ratio while keeping the graph connected. We can see that on average, KADER is able to lower the power assignment by 75%. The most important asset of a reduced power assignment is the increased battery lifespan. Yet, reducing the transmission range also causes less contention and interference for concurrent communications. This is even amplified for dense networks, where interferences reduce broadcasting efficiency. Therefore, KADER's low power assignment not only being energy efficient, also helps improve unicast and broadcast communications.

G.6 Convergence and Overhead Complexity of KADER's Topology

It is important to compute the communication complexity of KADER for topology creation. The communication complexity describes the average number of messages required to perform a protocol operation. Note that this comparison does not include the complexity of route discovery [10]. This issue is not covered in this chapter.

Message Complexity (MC)

In KADER, the network of n nodes is partitioned into m zones on the average, and each zone has an average number of nodes of $\frac{n}{m}$. The amount of communication overhead to build and maintain the forest is n since by sending n PN election messages, a forest will be constructed. To construct a zone, each node generates (d-1) messages to forward the learned PN or removed PN, where d is the hop-wise zone diameter. Therefore, each zone generates $(d-1)\frac{n}{m}$ messages. Since there exists m zones in the network, the overall number of generated forward messages becomes $(d-1)n$. In conclusion, the message complexity is O($\beta \cdot n \cdot d$),where $d < 3$ (see Figure III-18(c)).

Time Complexity (TC)

In order to obtain the time complexity induced by KADER, we need to analyze the broadcasting overhead in mobile ad-hoc network. Authors in [162] showed that Ω(n)

[10] KADER being zone-wise proactive, it is able to obtain intra-zone paths at no extra cost.

rounds are required by any broadcasting protocol when the network nodes are mobile. Accordingly, the broadcasting overhead does not change whether nodes are arranged in lines or in mesh, but only depends on the number of nodes in the network. Therefore, since KADER converges when a *Learn_PN* message has reached every node in a zone, and that KADER's zones have on the average n/m nodes, the time complexity of KADER is $\Omega(\mathrm{n/m})$.

G.7 Benefit of KADER's Topology on Routing Algorithms

KADER is able to derive the most stable links from a network topology such that full connectivity is always guaranteed. It then becomes interesting to analyze the benefits routing protocols may obtain from it.

Efficient Routing

KADER is able to group nodes into a set of zones, which proactively maintains routes between every node belonging to the same zone. Therefore, any routing protocol using KADER may perform *Intra_Zone* routing at no extra cost. For *Inter_Zone* routing, a protocol still needs to be determined. It could be imagined that a reactive approach, such that AODV [7] or DYMO [8], may take great help of the topology created in KADER by reducing the overhead of its *route discovery* procedure. This creates a hybrid routing protocol, using proactive intra-zone routing, and on-demand zone-level routing. On the other hand, similar to OLSR [4] using MPR, a proactive protocol takes benefit from KADER to improve its scalability and its end-to-end delay.

Energy Efficiency

In KADER, during the construction of the forest, every node elects its *Preferred Neighbor* partly depending on the energy needed to reach it. Indeed, the transmission range of the Intra_zone routing is always adapted to reach only the desired PN. Hence, by using the power assignment, KADER performs topology control and makes proactive *Intra_Zone* routes optimal in terms of energy data flow generated and forwarded by each node, further reducing the energy used for routing and increasing the channel capacity. It also improves concurrent communications by reducing interferences and contentions. Since KADER does not use beacons, routing protocols using KADER reach routing energy efficiency.

H CONCLUSION

In this chapter, we presented an original approach for applying mobility predictions to Mobile Ad Hoc Networks (MANET) called the Kinetic Graphs. The objective was to construct and maintain a topology or routing structure without relying on periodic maintenance. The approach is independent of the criteria used in order to build the backbone, and various approaches may be tested.

For example, we used the predicted nodes trajectory as the criterion for building a topology control protocol and developed KADER and showed how we could save energy and reduce interference.

However, KADER does not have totally local decisions, as some topology information has to be propagated in the trees. We therefore suggest as future work to adapt the same approach to the Localized Minimum Spanning Trees (LMST) algorithm and created the KLMST protocol. Due to the locality of all decisions, the KLMST protocol should actually correct coherency issues in KADER.

CHAPTER IV

Predictability of Mobility Models

Contents

Abstract— *In Palm Calculus, the Palm intensity of a particular transition is the model's expected number of transitions per time unit when considered at stationary regime. Considering transitions as the event of vehicles reaching a waypoint in MANETs' Mobility Models, the Inverse Palm Intensity (IPI) is defined as the mean interval between two waypoints, or the expected time spent by vehicles to reach a predefined target. This is also the predictability parameter we defined in Chapter I for measuring the efficiency of Kinetic Graphs. We propose in this Chapter to study this Palm intensity because such information is crucial in order to evaluate the predictability of MANETs Mobility Models. We obtain a lower bound for the IPI with reasonable configuration parameters situated at* $\approx$ **7s** *averaged on the Random-Waypoint mobility model (RWM) and* $\approx$ **9s** *for the City Section mobility model (CityM), both considered at steady state. Therefore, by wisely using this predictability interval, it is possible to adapt prediction models and improve the global performance of topology control and routing protocols.*

Keywords—*Predictability, Trajectory Duration, Mobility, Palm Calculus, Mobile Ad Hoc Networks.*

BEFORE designing a Mobility Prediction Algorithm, the first task is to derive a model that accurately describes the mobility pattern of the targeted system. The more complex the patterns are, the harder it is to generate prediction schemes. Once the model has been drawn, a key factor for the performance of prediction systems is directly related to the ability to extract future patterns based on past ones. In Information Theory, this feature is defined as the *mutual information* between past and future events, a low mutual information is called **uncertainty**, while a high mutual information is called **predictability**. It is therefore important to quantify the predictability of the mobility models in order to evaluate the potential gains the prediction schemes could provide.

A BASIC IDEA

Palm Calculus [163] is a set of formulae that relates time averages versus event averages. Time averages are obtained by sampling the system at arbitrary time instants. The event average viewpoint is obtained by sampling the system when selected state transitions occur. In MANETs, Palm Calculus is applied to mobility models in order to avoid subtle problems, such as speed decay of average speed as simulation progresses, or such as getting rid of differences between the long term distribution of nodes and the initial one. One important concept in Palm Calculus is the Palm Intensity. It is defined as the expected number of state transitions per time unit. When Palm calculus is applied to mobility models, a state transition is defined as the time instant when new parameters are set (direction, speed,..). The *Inverse Palm Intensity (IPI)* is therefore defined as the mean interval between two successive state transitions. Although the Palm distribution of speeds and positions of mobile nodes have already been asserted in [128], to our knowledge, the Palm intensity has never been analyzed in MANET's mobility models.

B SUMMARY OF CONTRIBUTION

In this Chapter, we make use of Palm Calculus to provide a theoretical lower bound on the mean interval between two successive waypoints, also called **trajectory duration**[1], for vehicular motion. We show that this value never falls below $7s$ on average for all reasonable practical purposes. This result is validated through simulations using the Random Waypoint Mobility model (RWM) and the City Section Mobility model (CityM) belonging to the Random Trip Framework [164]. It therefore motivates the use of mobility *reactive* strategies, since setting a lower bound on topology updates to 7 seconds makes the number of maintenance messages drop dramatically. Accordingly, it becomes conceivable to consider prediction-based models to reach optimal aperiodic maintenances.

[1]Therefore, Inverse Palm Intensity, Mean Interval between two successive waypoints, and Trajectory Duration will be later used interchangeably.

C ORGANIZATION OF WORK

The rest of the Chapter is organized as follows. In Section D, we develop a theoretical lower bound on the mean interval between two successive waypoints. Then, Section E shows simulation results and compares them with the theoretical approach. Finally, in Section F, we draw some concluding remarks and highlight future work.

D PALM INTENSITY

D.1 Random Waypoint

Palm calculus is now well established, but not widely used or even known in applied areas[2]. We do not use all the Palm Calculus framework here but only concentrate on the Palm Intensity λ. We apply Palm Calculus to the random waypoint model. We assume that this model has a stationary regime for a minimum velocity strictly greater than zero (see [165] for a complete proof) and consider as selected transitions instant T_n, the time at which waypoints are reached. Since the simulation is in stationary regime, we imagine that at time 0, the simulation has been running for some time. We take as convention $T_0 \leq 0 < T_1$. In other words, T_0 is the last time a transition occurred before time 0 and T_1 is the next one starting from 0. Considering $T_0 = 0$, the Palm intensity formula is given by

$$\frac{1}{\lambda} = \mathbb{E}^0(T_1 - T_0) = \mathbb{E}^0(T_1)$$

The inverse Palm intensity (IPI), or the mean interval between two successive waypoints, is therefore given by

$$\lambda^{-1} = \mathbb{E}^0(D_1)\mathbb{E}^0(\frac{1}{V_0}) = \overline{\Delta}\int_0^\infty \frac{1}{v} f^0(v)dv \tag{IV-1}$$

where D_1 and V_0 are two random variables representing nodes next position and speed respectively, and where $\overline{\Delta}$ is the average distance between two points in the simulation area and $f^0(v)$ is the Palm density distribution of speeds. The intensity is finite if and only if $\mathbb{E}^0(\frac{1}{V_0})$ is finite, which, for the uniform speed case, means $v_{min} > 0$. There exists a closed form for $\overline{\Delta}$ when the simulation area is a rectangle [166]. We consider here the simulation area as a $a \times a$ square and use the closed form $\overline{\Delta} \approx 0.5214a$.

D.2 Uniform Time Stationary Distribution of Speeds

When speeds are chosen from a uniform distribution with a low minimum speed, then at any given time, a large proportion of nodes will be moving very slowly. Since the average distance between nodes is fixed, this can create a nearly stable backbone that could make the Palm intensity seem unrealistically good. Therefore, the worst case for the intensity would be to have a uniform time stationary distribution of speeds which

[2]For a quick overview of Palm Calculus, refer to [165]; for a full fledged theory, see [163].

keeps vehicles velocities uniformly distributed through the simulation. Accordingly, the Palm intensity is reduced to the ratio between $\overline{\Delta}$ and the mean time stationary distribution of speeds. We also consider this case in our theoretical values and consider an appropriate choice of $f^0(v)$ proposed in [167]:

$$f^0(v) = \frac{2s}{v_{max}^2 - v_{min}^2} \text{ for } v_{min} < s < v_{max} \tag{IV-2}$$

Therefore, the inverse intensity is reduced to

$$\lambda^{-1} = \overline{\Delta}\frac{2}{v_{max} + v_{min}} \tag{IV-3}$$

D.3 Random Waypoint with Pausing

When a mobile reaches a waypoint, it picks a pausing duration according to the density f_{pa}^0, stays immobile for this duration, and then moves again. To analyze this model, we consider as selected transition times the time instants at which either a waypoint is reached or a pausing time expires. From [165], the intensity formula gives

$$\frac{1}{\lambda} = 0.5\frac{1}{\lambda_{pa}} + 0.5\frac{1}{\lambda_{mo}} \tag{IV-4}$$

D.4 City Section

A final consideration would be the analysis of the theoretical IPI for the City Section mobility model (CityM). Since it is a special case of the random waypoint model on a non-convex domain, (IV-1) or (IV-3) may be applied. The domain is the union of the segments defined by the edges of a space graph (Fig. IV-6(a) for example). Therefore, $\overline{\Delta}$ is the average distance between two segments in the domain, which is specific to each map. For example, in Fig. IV-6(a), $\overline{\Delta} \approx 163m$.

E EXPERIMENTAL RESULTS

We assessed the Inverse Palm intensity through simulations and compared it with the theoretical values obtained in the previous section (see solid lines in the following figures). We have evaluated the IPI for both the RWM and the CityM in which we have used different real topological maps. Since both mobility models are extracted from the Random Trip Framework [164], the IPIs are obtained when both models are at steady state.

The following figures show the characteristics of the Inverse Palm Intensity (IPI) given the mean speed on the RWM and CityM evaluated at regular $5m/s$ intervals. The solid line represents the theoretical value for the IPI, while the boxes are the experimental ones. Each box represents 10 runs. We made node average velocities v^{av} vary from 5 to 50 $[m/s]$, yet with different variances α around the mean speed. In other words, a node velocity is uniformly distributed between $[\mathrm{v}^{\mathrm{av}} - \alpha, \mathrm{v}^{\mathrm{av}} + \alpha]$. We have simulated

$900s$ of both models for three different pausing times: 2, 5, 10 $[m/s]$. Nodes were assumed to be moving in a flat squared area of $1000m \times 1000m$. Finally, we defined $\mathrm{IPI}^{\mathrm{uni}}$ as the IPI obtained by simulating the RWM or the CityM using the uniform time stationary distribution of speeds given by (IV-2).

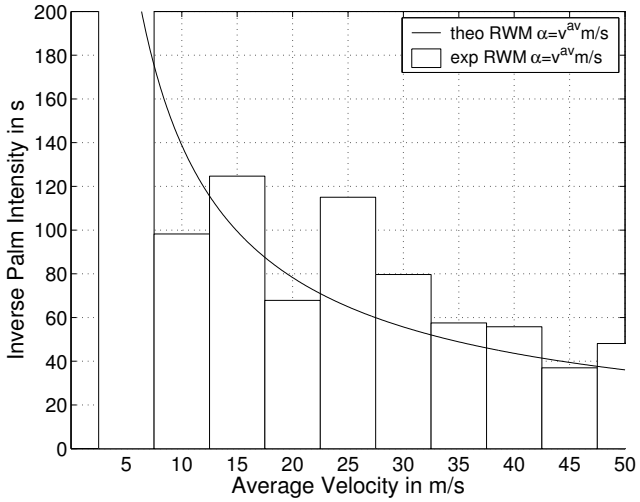


Fig. IV-1. Comparison of the experimental and theoretical IPI for the RWM at stationary regime

Fig. IV-1 illustrates the theoretical and the experimental IPI of the RWM considered at stationary regime. Similarly, Fig. IV-2 depicts the theoretical and the experimental $\mathrm{IPI}^{\mathrm{uni}}$ of the RWM also at stationary regime. According to those two Figures, the IPI and $\mathrm{IPI}^{\mathrm{uni}}$'s lowest bound for the RWM are $\approx 40s$ and $\approx 10s$ respectively. Yet, when considering moderate values for the average velocity such as $25m/s$, we can see that the IPI is $\approx 60s$ in Fig. IV-1 and $\approx 20s$ in Fig. IV-2. As explained in Section D.2, Fig. IV-2 is as expected the worst case configuration for the RWM given its uniform time-stationary distribution of speeds. Indeed, the IPI values are on average 75% smaller than IPI obtained with a non-time-stationary distribution.

Fig. IV-3 depicts the evolution of the IPI of the RWM when the degree of liberty around the average velocity, α, is increased. We can see that the IPI increases when we increase α from 5, 10, and 15 m/s. This may be explained by the fact that since the average distance between nodes is fixed, the IPI is only influenced by α in (IV-1). This remark is also corroborated by the large values obtained for the IPI in Fig. IV-1, where α is maximized to v^{av}. Then, considering (IV-3), α does not have any influence on the $\mathrm{IPI}^{\mathrm{uni}}$. Therefore, varying α on the RWM will only influence simulations using non uniform time-stationary distribution of speeds. And as it may be seen when comparing Fig. IV-3 and Fig. IV-2, increasing α will only increase the gap between IPI and $\mathrm{IPI}^{\mathrm{uni}}$. This is therefore the final belief when considering uniform time-stationary distribution of speed as the most restricting configuration parameter for the IPI. As soon as we relax this constraint and increase α, the IPI becomes more attractive.

We then show in Fig. IV-4 the influence of the pause times on the IPI for the RWM. In

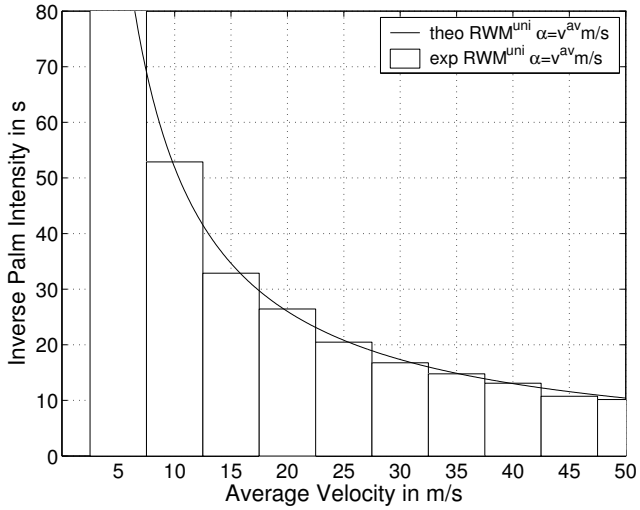
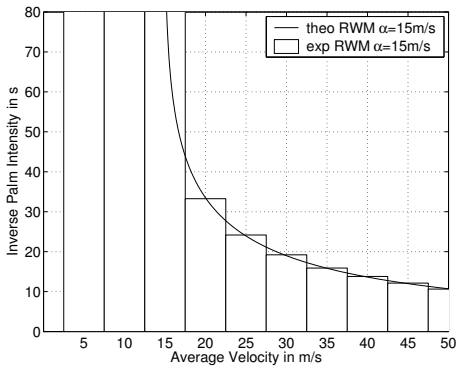


Fig. IV-2. Comparison of the experimental and theoretical $\mathrm{IPI}^{\mathrm{uni}}$ of the RWM at stationary regime with a uniform time stationary distribution of speeds

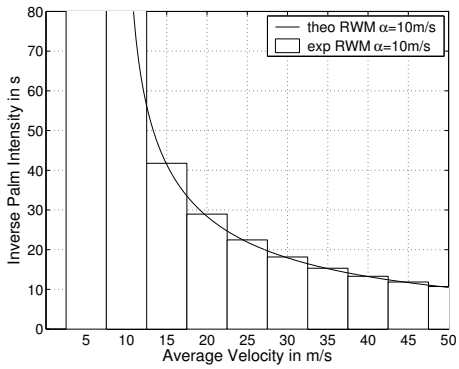
this figure, we can see that the RWM with pausing gives smaller values for the IPI than all other configurations for the RWM due to the IPI of the pause transitions (Eq. IV-4). This can be explained by the fact that when RWM is simulated with pause times, the event *leave a waypoint* needs to be considered as a second transition case, hence increasing the average transitions per unit of time. As we could expect, the smallest IPIs are obtained for large velocities. When considering the RWM with small pausing times, the IPI is $\approx 7s$ (see Fig. IV-4(a)). Yet, it improves when the average pausing time increases. In Fig. IV-4(b), the IPI $\approx 8s$ and in Fig. IV-4(c), it is $\approx 12s$. As before, for more reasonable velocities, the IPI becomes more attractive.

Finally, we simulated the City section mobility model (CityM) in Fig. IV-5, IV-6, and IV-7 with different road topologies extracted from the US Census Bureau TIGER database [105]. Vehicles are moving at constant speed on a road and each intersection represents a waypoint. Since the mean distance between intersections is far lower in our maps than in the RWM simulations, we obtained smaller values for the IPI than those in the RWM. We only considered here a range of average velocities varying up to $20m/s$, since larger velocities would not be acceptable for realistic situations. The simulation parameters are similar to those of the RWM model. As we mentioned before, uniform time-stationary distribution of speeds is the most restrictive configuration for the IPI. Since we wish to obtain a lower bound for the IPI, we assumed nodes velocities to be uniform time-stationary and we accordingly used (IV-3) in order to obtain the theoretical $\mathrm{IPI}^{\mathrm{uni}}$.

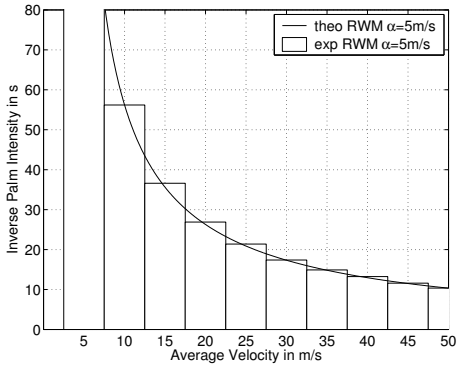
In Fig. IV-5, we simulated the CityM on two different highly dense urban areas, while in Fig. IV-6 we depicted the CityM on two lightly dense urban areas. We thought that instead of randomly using a large number of different maps, it would be more interesting to find different classes of topological maps and compute the IPI accordingly. We can see that for fairly large velocities, the IPI does not vary that much. In Fig-



(a) α = 15 m/s.

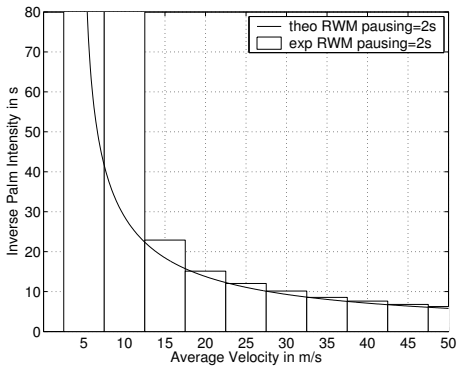


(b) α = 10 m/s.

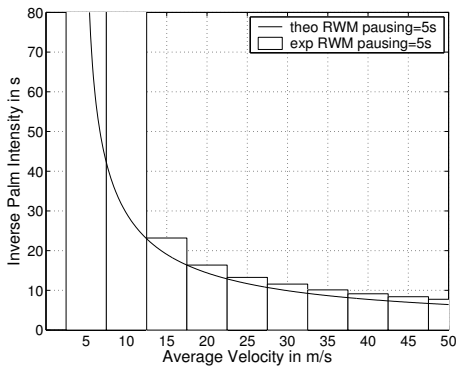


(c) α = 5 m/s.

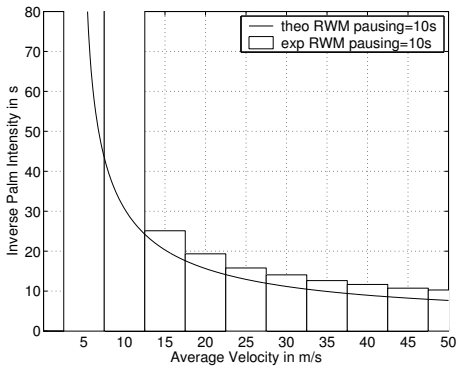
Fig. IV-3. Comparison of the experimental and theoretical IPI of the RWM at stationary regime for two different values of α.



(a) pause time = 2s.

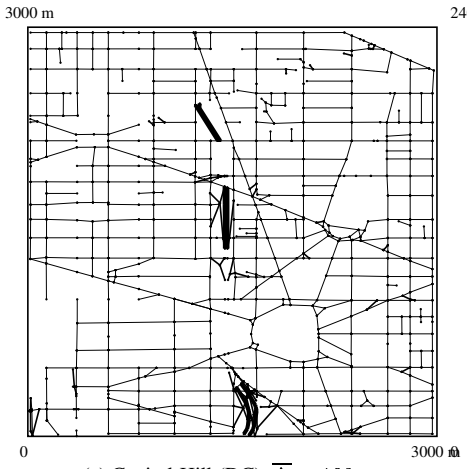


(b) pause time = 5s.

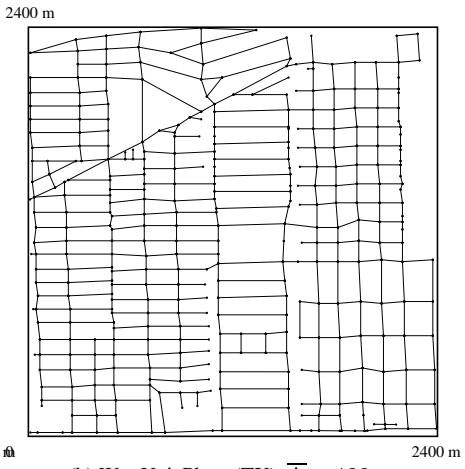


(c) pause time = 10s.

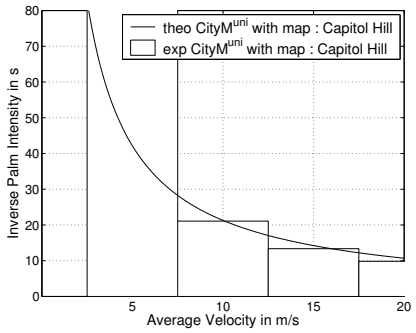
Fig. IV-4. Comparison of the experimental and theoretical IPI for the RWM at stationary regime for different values of pausing times.



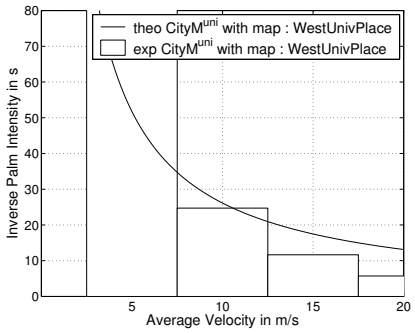
(a) Capitol Hill (DC), $\overline{\Delta} \approx 108m$



(b) WestUnivPlace (TX), $\overline{\Delta} \approx 132m$

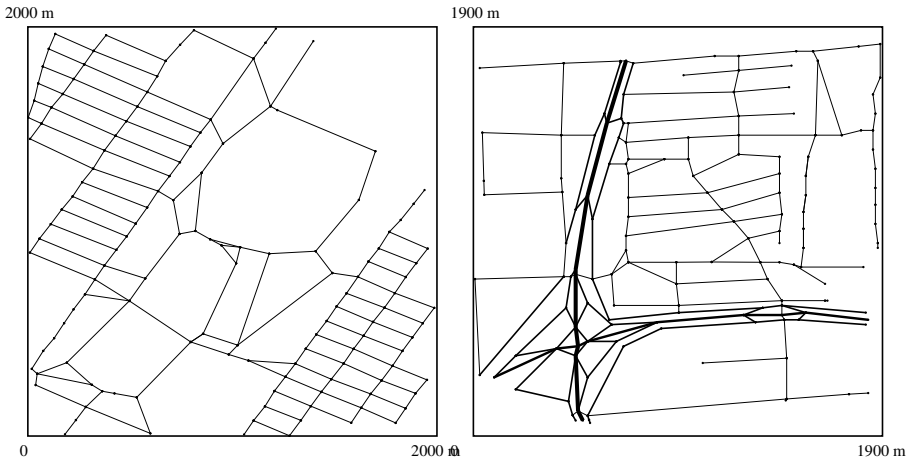


(c) Inverse Palm intensity for the map of Capitol Hill



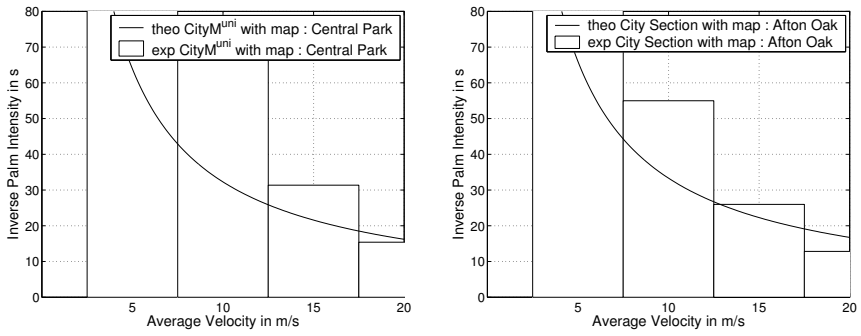
(d) Inverse Palm intensity for the map of WestUnivPlace

Fig. IV-5. Comparison of the experimental and theoretical IPI for the CityM in two highly dense urban areas



(a) South Central Park (NYC), $\overline{\Delta} \approx 163m$

(b) Afton Oak (TX), $\overline{\Delta} \approx 168m$



(c) Inverse Palm intensity for the map of South Central Park

(d) Inverse Palm intensity for the map of Afton Oak

Fig. IV-6. Comparison of the experimental and theoretical IPI for the CityM in two lightly dense urban areas

ures IV-5(c) and IV-5(d), IPI $\approx 9s$, while in Figures IV-6(c), and IV-6(d), IPI $\approx 13s$. It is rather for smaller velocities that the IPIs differs more dramatically. Indeed, when nodes are moving at $10m/s$, the IPI in Figures IV-5(c) and IV-5(d) is $\approx 23s$, while in Figures IV-6(c), and IV-6(d), the IPI $\approx 60s$.

Another observation is that for lightly dense urban areas, and particularly for low mean speeds, the theoretical $\mathrm{IPI}^{\mathrm{uni}}$ differs from the experimental one. Indeed, as it may be observed in Fig. IV-5 or in Fig. IV-6 for larger mean speeds, the practical and theoretical $\mathrm{IPI}^{\mathrm{uni}}$ are more correlated. This particular effect may be explained by the fact that unlike the RWM, the path between an initial point and a target point in the CityM is not a straight line, but is rather determined using Dijkstra's shortest path algorithm applied to the map. Yet, this feature is neither taken into account by (IV-1) nor by (IV-3). Therefore, for the same reasons the IPI differs from $\mathrm{IPI}^{\mathrm{uni}}$ for the RWM, the experimental $\mathrm{IPI}^{\mathrm{uni}}$ will observe less stability around the theoretical value. And this feature is more probably to appear in lightly dense areas, or for low mean speeds because in those two cases, the ratio $\mathrm{segment\_length/speed}$ makes nodes more sensitive to the choice of the segments by Dijkstra's algorithm.

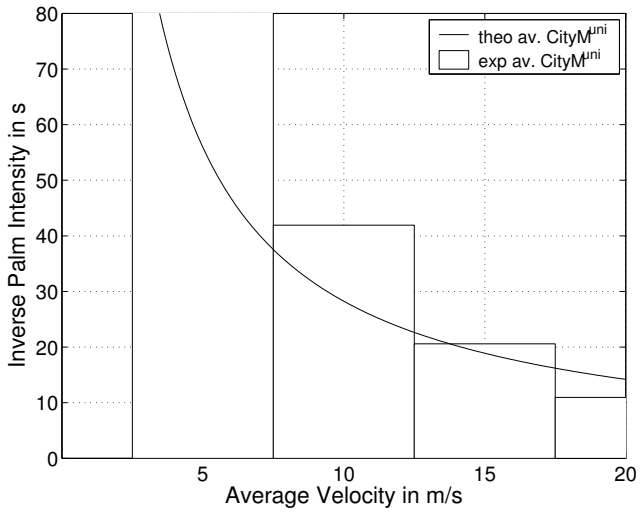


Fig. IV-7. Comparison of the experimental and theoretical average IPI for the CityM

We finally averaged the IPI of our maps in Fig. IV-7. This figure represents the average theoretical and practical IPI of the City Section mobility model. According to this figure, when the mean speed is $20m/s$, the IPI is $\approx 10s$. But when the mean speed is smaller, we can expect a much larger IPI. Indeed, for a mean speed of $10m/s$, IPI $\approx 40s$, which is corroborated by the theoretical IPI. Moreover, in a configuration where mean speeds could vary depending on the street category (similar to a speed limit for example), the IPI would be even bigger than that.

F CONCLUSION

We have provided a theoretical and experimental lower bound on the average trajectory duration, or Inverse Palm Intensity (IPI), that is $\approx 7s$ and $\approx 9s$ using extreme values for the configuration parameters of the Random Waypoint mobility model and the City Section mobility model respectively. We also illustrated that the values obtained by the Palm Calculus in Section D modeled the practical ones correctly, and could be directly used in order to extract trajectory durations from various mobility models. Finally, we pointed out that for realistic situations, the trajectory duration observed by mobile nodes is $\approx 30s$.

This outcome is interesting as it provides a lower bound for the predictability, an conversely an upper bound on the uncertainty, of mobility patterns generated by random mobility models for MANETs. It therefore motivates the use of prediction techniques and Kinetic Graphs in order to exploit this precious information in order to optimize the refreshing processes of topology control and routing protocols. For example, considering the IETF recommendations for operating OLSR [4], if we set the topology update intervals to $10s$, the corresponding overhead could be reduced by up to 85% .

Although very interesting, these values depend on nodes' average velocity and on the distance between two successive waypoints. Even though it is not an easy task to obtain a good estimate of their values in real situations, we can find a dual behavior for pedestrian and vehicular motions. When nodes move fast, they usually follow predefined routes and their trajectories may be easily predicted. But when nodes experience random walks, they usually move at a lower speed and results obtained in this Chapter give estimates on their average trajectory duration. Therefore, nodes mobility assessment depends on the application for the deployment of mobile ad-hoc networks.

An interesting future orientation is to perform a thorough study on trajectory lengths in real environments using mobility traces or realistic mobility models. Indeed, the mobility patterns might not show a similar regularity than those of the random trip framework. Now that we have obtained an insight of the average trajectory durations in random mobility models, it would be interesting to see if we could obtain similar values in real deployments. For that matter, we would need to think of a more realistic mobility model fitting real patterns, and devise more complex prediction schemes in order to reduce the uncertainty of the generated motion patterns.

CHAPTER V

Modeling Vehicular Mobility Patterns

Contents

Abstract— *Although having been used in various fields of mobile ad hoc networking in past years, the Random Waypoint Model, and Random Models in general, have attracted criticisms for their lack of realistic motion modeling for specific MANET applications. Among them, inter-vehicular communications (IVC), which require an accurate modeling of vehicular motion patterns, are attracting a growing attention from both academia and industry, due to the amount and importance of the related applications, ranging from road safety, traffic control, and mobile entertainment. Although faithful vehicular mobility modeling is a challenging field, vehicles usually show some signs of regularities in their patterns which could be therefore exploited by prediction techniques.*

Because of their peculiar characteristics, IVCs require the definition of specific networking techniques, whose feasibility and performance are usually tested by means of simulation. One of the main challenges posed by simulations for IVCs is the faithful characterization of vehicular mobility at both macroscopic and microscopic levels, leading to realistic non-uniform distributions of cars and velocity, and unique connectivity dynamics. Yet, freely distributed tools which are commonly used for academic studies only consider limited vehicular macro-mobility issues, while they pay little or no attention to vehicular micro-mobility and its interaction with the macro-mobility counterpart. Such a simplistic approach can easily raise doubts on the confidence of derived IVCs simulation results. In this Chapter we present and describe VanetMobiSim, a freely available generator of realistic vehicular movement traces for telecommunication networks simulators. VanetMobiSim is validated first by illustrating how the interaction between featured macro- and micro-mobility is able to reproduce typical phenomena of vehicular traffic. Then, the traces generated by VanetMobiSim are formally validated against those obtained from CORSIM, a benchmark traffic generator in transportation research.

Keywords—*Traffic Generator, Vehicular Mobility Patterns, Simulator, Validation, Vehicular Ad Hoc Networks.*

IN the previous chapter, we provided an insight on the predictability of the Random Waypoint Mobility (RWM) model. However, this model recently attracted a lot of criticism for the lack of realism of the generated mobility patterns. Indeed, it cannot faithfully model constrained motions, which are common in almost all applications of mobile networks. For example, vehicular mobility typically involves complex spatial and temporal dependencies which cannot be addressed by the RWM. Vehicular motions moreover seem to show some kind of regularities which could be exploited by appropriate prediction schemes. However, a prediction algorithm is strongly interlinked with the mobility model generating the patterns it aims at predicting.

A MOTIVATION

Vehicular Ad-hoc Networks (VANETs) represent a rapidly emerging, particularly challenging class of Mobile Ad Hoc Networks (MANETs) to be used for Inter-Vehicular Communications (IVC). VANETs are distributed, self-organizing communication networks built up from traveling vehicles, and are thus characterized by very high speed and limited degrees of freedom in nodes movement patterns. Such particular features often make standard networking protocols inefficient or unusable in VANETs, and this, combined with the huge impact that the deployment of VANET technologies could have on the automotive market, explains the growing effort in the development of communication protocols which are specific to vehicular networks.

Whereas it is crucial to test and evaluate protocol implementations in real testbed environments, logistic difficulties, economic issues and technology limitations make simulation the mean of choice in the validation of networking protocols for VANETs, and a widely adopted first step in development of real world technologies. A critical aspect in a simulation study of VANETs, is the need for a mobility model which reflects, as close as possible, the real behavior of vehicular traffic. When dealing with vehicular mobility modeling, we distinguish between macro-mobility and micro-mobility descriptions.

For *macro-mobility*, we intend all the macroscopic aspects which influence vehicular

traffic: the road topology, constraining cars movement; the per-road characterization, defining speed limits, number of lanes, overtaking and safety rules over each street of the aforementioned topology; the traffic signs description, establishing the intersections crossing rules; the car class dependent constrains, providing differentiation in the above rulings for different types of vehicles; the traffic patterns delineation, outlining the popularity of different locations as traffic destinations during different hours of the day and for different classes of drivers, etc.

Micro-mobility instead refers to the individual behavior of drivers, when interacting with other drivers or with the road infrastructure: traveling speed in different traffic conditions; acceleration, deceleration and overtaking criteria; conduct in presence of road intersections and traffic signs; general driving attitude, related to driver's age, sex and mood, etc. The distinction between macro- and micro-mobility we propose is not to be confused with the difference between the macroscopic and microscopic scales commonly employed in traffic flow theory, and in physics in general. In that contest, macroscopic descriptions model gross quantities of interest, such as density or mean velocity of cars, treating vehicular traffic according to fluid dynamics, while microscopic descriptions consider each vehicle as a distinct entity, modeling its behavior in a more precise but computationally more expensive way.

It would be desirable for a trustworthy VANETs simulation that both macro-mobility and micro-mobility descriptions be jointly considered in modeling vehicular movements. Indeed, many MANET mobility models employed in VANETs simulations ignore these guidelines, and thus fail to reproduce peculiar aspects of vehicular motion, such as car acceleration and deceleration in presence of nearby vehicles, queuing at road intersections, clustering caused by semaphores, vehicular congestion and traffic jams.

B Summary of Contribution

In this Chapter, we introduce VanetMobiSim [78], a freely distributed, open source vehicular mobility generator based on the CanuMobiSim architecture [116] and designed for integration with telecommunication network simulators. VanetMobiSim can produce detailed vehicular movement traces employing different macro- and micro-mobility models and taking into account the interaction of the two, and can simulate different traffic conditions through fully customizable scenarios. We validate the mobility patterns generated by VanetMobiSim by recreating distinctive vehicular mobility effects, such as speed decay with increasing car density, non-uniform distribution of vehicles in urban areas, and shock waves due to stop-and-go perturbations. We also formally validate VanetMobiSim by comparing this vehicular traces with those generated by a benchmark traffic generator in the transportation community.

C Organization of Work

The rest of the Chapter is organized as follows. Section D illustrates the need for realistic simulations in vehicular networks. A detailed description of the features of VanetMobiSim is given in Section E. Section F presents validating tests on movement

traces produced by VanetMobiSim in specific scenarios and by comparison with TSIS-CORSIM. Finally, in Section G, we discuss the outcome of this chapter and outline future research directions.

D THE NEED FOR REALISM IN VEHICULAR TRAFFIC MODELING

Only in recent times has networking community started paying attention to the impact that realistic mobility modeling has on vehicular communications.

The use of simplistic mobility models that has characterized most of the literature on the topics of mobile and vehicular networks appears an evident flaw, when considering that vehicular traffic theory has undergone fifty years of increasingly accurate studies.

When comparing mobility models employed in recent works on vehicular networks and analytical descriptions following well known approaches of traffic theory, the difference in terms of results is dramatic, and it is clear that such a discrepance cannot have a null impact on the performance of networking protocols and techniques.

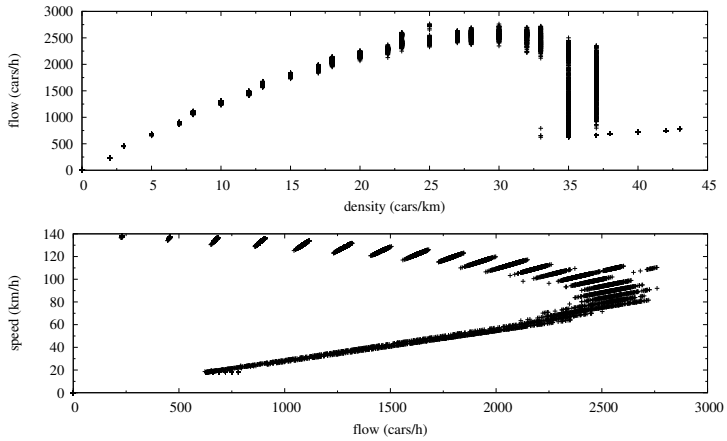


Fig. V-1. Flow versus density and speed versus flow under the Fluid Traffic Model

In traffic theory, since the 60's, models reproducing drivers behavior have been subject to standard tests in order to be considered realistic enough. As an example, a minimal requirement is a model capable of recreating the lambda-shaped relation between vehicular flow and density. Even low-complexity traffic stream models can reproduce it, as they look at vehicular mobility as a hydrodynamic phenomenon, and thus do not model the behavior of each car individually. An example is shown in Fig. V-1, which is depicting both the aforementioned lambda-shaped relation and the one obtained for the speed and flow when using the Fluid Traffic Model implemented in VanetMobiSim. The reason at the basis of the phenomena is that, given a straight road, the out-flow of vehicles grows linearly at first, as the in-flow rate, and consequently the car density, is increased. However, when the critical vehicular density is reached, the road capacity cannot sustain the arrival rate anymore, leading to queueing phenomena that slow down the system as the density increases.

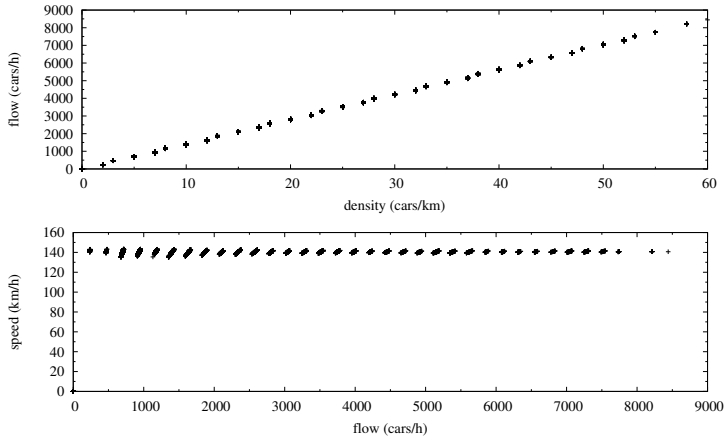


Fig. V-2. Flow versus density and speed versus flow under the Manhattan Model

When the same test is performed on the previously mentioned Manhattan model, the results depicted in Fig. V-2 are not matching the expectations. Even if the Manhattan model implements some bounded randomness in the velocity update, and imposes speed limitations to avoid overlapping of vehicles, the lack of a desired speed and of accurate car following rules make the description unrealistic as the growth in the in-flow is producing a linear increase on the car density.

Speed waves represent another condition of vehicular traffic commonly reproduced during the validation process of a mobility model in traffic theory works. These perturbations are known to be generated by heavy traffic conditions on highways or by periodic obstacles such as traffic lights or entering ramps, and are due to the finite response time of drivers to slowdowns determined by such events. As depicted in Fig. V-3, where slow speed dark waves move against the direction of traffic in time, a car following model like the Intelligent Driver Model (IDM) implemented in VanetMobiSim and discussed later in the chapter, can correctly recreate this phenomenon. The equivalent plot obtained using the Manhattan model appears as a white image, since all the vehicles maintain the maximum speed, and is thus not shown here. As shown in Fig. V-3, the Fluid Traffic Model fails to reproduce the desired behavior in this case, since this model does not include a car-to-car interaction description.

Another typical proof of the validity of a vehicular mobility model is determined by its response to dynamic situations, such as that occurring to a queue of cars in presence of an obstacle ahead suddenly removed. In that case, it is expected that the model forces the drivers to slow down while approaching the obstacle and then to accelerate again once the impediment is removed. This is actually what we can observe in Fig. V-4 when IDM is used. Each line represents the evolution of speed over time of one car and for the first twenty vehicles in the queue. It can be noticed that the first vehicle slows down as the obstacle becomes nearer, and that the cars behind follow the leader's speed dynamics with some delay due to the drivers' reaction time. When the obstacle is removed, just before the leading car stops completely, the vehicles start accelerating again towards full speed. The cars back in the queue experience a different speed evolution, as they are far from the obstacle and are thus still moving at high speed when the impediment is removed. The same is not true when the Manhattan model

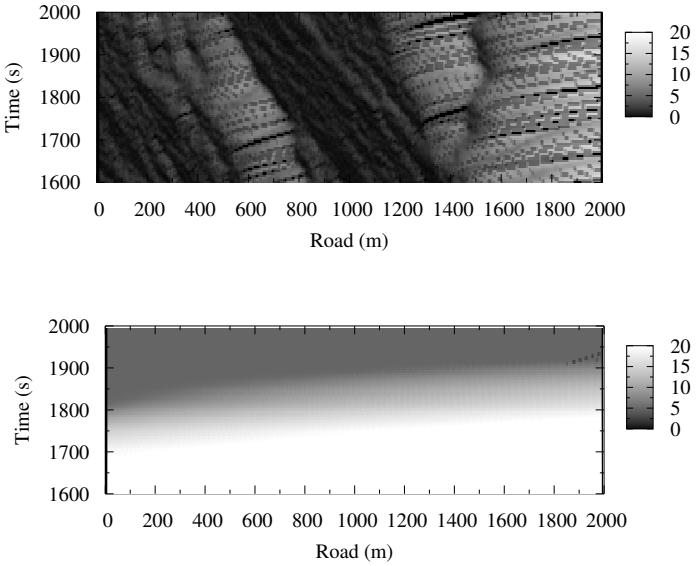


Fig. V-3. Speed versus time and space in a highway scenario, in presence of increasing car in-flow, when using the Intelligent Driver Model (upper) and the Fluid Traffic Model (bottom)

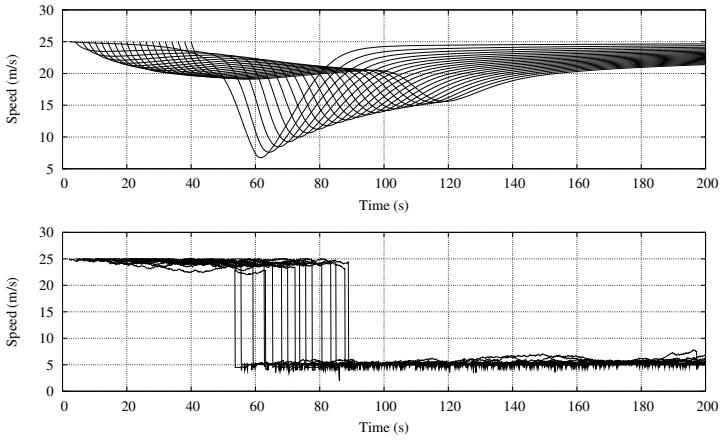
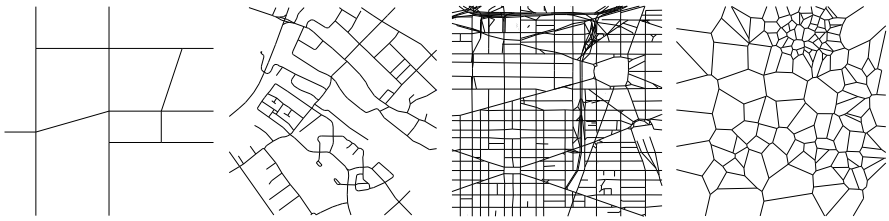


Fig. V-4. Evolution of speed for the first 20 vehicles belonging to a queue of cars meeting an obstacle which is then suddenly removed. The plots refer to the case in which the Intelligent Driver Model is employed (upper) and that in which the Manhattan model is used (bottom)

is used, as the model prevents vehicle overlapping by abruptly reducing to zero the speed of the leading vehicle when it reaches the obstacle. Furthermore, it is not able to induce a free-flow acceleration due to the lack of a desired speed description. The cars in the queue are forced to strictly follow the leading vehicle behavior, and thus describe similar curves. The resulting plot is shown in Fig. V-4.

E VANETMOBISIM

VanetMobiSim is an extension to CanuMobiSim [116], a generic *user mobility* simulator. CanuMobiSim is a platform- and simulator-independent software coded in *Java*, and producing mobility traces for different network simulators, including *ns-2* [130], *GloMoSim* [131], *QualNet* [132] and *OPNET* [133]. It provides an easily extensible mobility architecture, but, due to its general purpose nature, suffers from a reduced level of detail in specific scenarios. VanetMobiSim is therefore aimed at extending the vehicular mobility support of CanuMobiSim to a higher degree of realism. In this section, we outline the structure and characteristics of VanetMobiSim and detail the resulting vehicular mobility support.



(a) User-defined topology (b) GDF map topology (c) TIGER map topology (d) Clustered Voronoi

Fig. V-5. Road topologies examples

E.1 Macro-mobility Features

When considering macro-mobility, we not only take into account the road topology, but also the road structure (unidirectional or bidirectional, single- or multi-lane), the road characteristics (speed limits, vehicle-class based restrictions) and the presence of traffic signs (stop signs, traffic lights, etc.). Moreover, the concept of macro-mobility also includes the effects of the presence of points of interests, which influence vehicles movement patterns on the road topology. All these different aspects of the macro-mobility are discussed in details in the remainder of this section.

Road topology definition

The selection of the road topology is a key factor for obtaining realistic results when simulating vehicular movements. Indeed, the length of the streets, the frequency of

intersections, or the density of buildings can greatly affect important mobility metrics such as the minimum, maximum and average speed of cars, or their density over the simulated map. VanetMobiSim allows to define the road topology in the following ways:

- *User-defined graph*: the road topology is specified by listing the vertices of the graph and their interconnecting edges.
- *GDF map*: the road topology is imported from a Geographical Data File (GDF) [117]. Unfortunately, most GDF file libraries are not freely accessible.
- *TIGER map*: the road topology is extracted from a map obtained from the TIGER database [105]. The level of detail of the maps in the TIGER database is not as high as that provided by the GDF standard, but this database is open and contains digital descriptions of wide urban and rural areas of all districts of the United States. In fact, topology descriptions from the TIGER database are becoming quite common in VANETs simulation.
- *Clustered Voronoi graph*: the road topology is randomly generated by creating a Voronoi tessellation on a set of non-uniformly distributed points. This approach is similar to that proposed in [168], but we also consider the presence of areas with different road densities which we refer to as *clusters*. The number of clusters and their density are customizable to represent diverse geographical characterizations in the same map, such as city centers, suburban areas, or countryside. The clustered Voronoi graph can be especially useful to rapidly generate large road topologies.

In all these cases, the road topology is implemented as a graph over whose edges the movement of vehicles is constrained. Examples of different VanetMobiSim topologies are illustrated in Fig. V-5.

Road topology characterization

As stated before, the concept of vehicular macro-mobility is not limited to motion constraints obtained from graph-based mobility, but also includes all aspects related to the road structure characterization, such as directional traffic flows or multiple lanes, speed constraints or intersection crossing rules. None of these aspects is present in CanuMobiSim, thus the following enhancements are introduced by VanetMobiSim:

- introduction of roads with multiple lanes in each direction
- physical separation of opposite traffic flows on each road
- definition of independent speed limits on each road of the topology
- implementation of traffic signs at each road intersection. By default, intersections are fully regulated by stop signs. Alternatively, it is possible to regulate traffic at intersections by means of traffic lights.

Note that, for the road topology characterization to have an impact on vehicular mobility, a strong interaction between the macro-mobility description and the micro-mobility models that define drivers behavior is required. Thus, the micro-mobility model must be designed to keep roads characteristics in consideration. This issue is discussed in Section E.2.

Vehicular movement patterns selection

Vehicular traffic schemes in urban scenarios are far from being random. Indeed, cars tend to move between points of interests, which are often common to many drivers and can change in time (e.g., offices may be strong attraction points, but mainly during the first part of the morning). Accordingly, VanetMobiSim exploits CanuMobiSim capability of building up movement patterns from the cooperation of a *trip generation module*, which defines the sets of points of interest, and a *path computation module*, whose task is to compute the best path between those points.

Two choices are given for the trip generation module. The first is a *random trip*, as the start and stop points of movement patterns are randomly selected among the vertices of the graph representing the road topology. The second is an *activity sequences* generation, in which a set of start and stop points are explicitly provided in the road topology description, and cars are forced to move among them. In particular, multiple sets of points of interest can be specified, along with the probability matrix of a vehicle switching from one set to another.

Independently from the trip generation method employed, the path computation, i.e. the selection of the best sequence of edges to reach the selected destination, can be performed in three ways. The first method selects the shortest path to destination, running a *Dijkstra's algorithm* with edges cost inversely proportional to their length. The second method does not only considers the length of the path, but also the *traffic congestion level*, by weighting the cost of traversing an edge also on the number of cars traveling on it, thus modeling the real world tendency of drivers to avoid crowded paths. The last method, which is not present in the original CanuMobiSim, extends the other two, by also accounting for the *road speed limit* when calculating the cost of an edge, in a way that fastest routes are preferred.

The combination of trip generation and path computation methods offers a wide range of possibilities, when the definition of vehicular movement paths is a factor of interest in the mobility simulation. The best practice depends on the application. But generally, the mobility patterns are more realistic if we use similar criteria as real drivers, in other words, if we use an *activity sequence* trip generation in conjunction with a speed path selection.

E.2 Micro-Mobility Features

The concept of vehicular micro-mobility includes all aspects related to an individual car's speed and acceleration modeling. The micro-mobility description plays the main role in the generation of realistic vehicular movements, as it is responsible for effects such as smooth speed variation, cars queues, traffic jams and overtakings.

Three broad classes of micro-mobility models, featuring an increasing degree of detail, can be identified depending on whether the individual speed of vehicles is computed i) in a deterministic way, ii) as a function of nearby vehicles behavior in a single lane scenario, or iii) as a function of nearby vehicles behavior in a multi-flow interaction (i.e., urban) scenario.

CanuMobiSim provides implementations for models belonging to the first two classes. The *Graph-Based Mobility Model* (GBMM) [169], the *Constant Speed Motion* (CSM) [116] and the *Smooth Motion Model* (SMM) [170] fall into the first category, as the speed of each vehicle is determined on the basis of the local state of each car and any external effect is ignored. They all constrain a random movement of nodes on a graph, possibly including pauses at intersections (CSM) or smooth speed changes when reaching or leaving a destination (SSM). The movement is random in a sense that vehicles select one destination and move towards it along a shortest-length path, ignoring (and thus possibly overlapping) other vehicles during the motion. While these models may work for isolated cars, they fail to reproduce realistic movements of groups of vehicles.

The *Fluid Traffic Model* (FTM) [171] and *Intelligent Driver Model* (IDM) [97] are instead part of the second class, as they account for the presence of nearby vehicles when calculating the speed of a car. These models describe car mobility on single lanes, but do not consider the case in which multiple vehicular flows have to interact, as in presence of intersections.

The FTM describes the speed as a monotonically decreasing function of the vehicular density, forcing a lower bound on speed when the traffic congestion reaches a critical state according to the following equation

$$s = max\left[s_{min}, s_{max}\left(1 - \frac{k}{k_{jam}}\right)\right]$$

where s is the output speed, s_{min} and s_{max} are the minimum and maximum speed respectively, k_{jam} is the vehicular density for which a traffic jam is detected, and k is the current vehicular density of the road the node is moving on. This last parameter is given by $k = n/l$, where n is the number of cars on the road and l is the length of the road segment itself. According to this model, cars traveling on very crowded and/or very short streets are forced to slow down, possibly to the minimum speed, if the vehicular density is found to be higher than or equal to the traffic jam density. On the other hand, as less congested and/or longer roads are encountered, the speed of cars is increased towards the maximum speed value. Thus, the Fluid Traffic Model describes traffic congestion scenarios, but still cannot recreate queuing situations, nor can it correctly manage cars behavior in presence of road intersections. Moreover, no acceleration is considered and it can happen that a very fast vehicle enters a short/congested edge, suddenly changing its speed to a very low value, which is definitely a very unrealistic situation.

On the other hand, the IDM characterizes drivers behavior depending on their front vehicle, thus falling into the so-called *car following* models category. The instantaneous acceleration of a vehicle is computed according to the following equations

$$\frac{dv}{dt} = a\left[1 - \left(\frac{v}{v_0}\right)^4 - \left(\frac{s^*}{s}\right)^2\right] \qquad \text{and} \qquad s^* = s_0 + \left(vT + \frac{v\Delta v}{2\sqrt{ab}}\right)$$

In the left hand Equation, v is the current speed of the vehicle, v_0 is the desired velocity, s is the distance from preceding vehicle and s^* is the so called *desired dynamical distance*. This last parameter is computed as shown in the right hand equation, and is a function of the minimum bumper-to-bumper distance s_0, the minimum safe time headway T, the speed difference with respect to front vehicle velocity Δv, and the maximum acceleration and deceleration values a and b. When combined, these formulae give the instantaneous acceleration of the car, divided into a "desired" acceleration $[1-(v/v_0)^4]$ on a free road, and braking decelerations induced by the preceding vehicle $(s^*/s)^2$.

VanetMobiSim adds two original microscopic mobility models, both of which account for the interaction of multiple converging flows by acting consistently with the road infrastructure, and thus fall into the third category mentioned above. These models extend the IDM description, which is the most realistic among those present in CanuMobiSim, in order to include the management of intersections regulated by traffic signs and of roads with multiple lanes. We also would like to emphasize that, as both models extends IDM, they are also able to reproduce a lambda-shape relation between vehicular flow and density.

The first new micro-mobility model is referred to as *Intelligent Driver Model with Intersection Management* (IDM-IM). It adds intersection handling capabilities to the behavior of vehicles driven by the IDM. In particular, IDM-IM models two different intersection scenarios: a crossroad regulated by stop signs, or a road junction ruled by traffic lights. In both cases, IDM-IM only acts on the first vehicle on each road, as IDM automatically adapts the behavior of cars following the leading one. Every time a vehicle finds no intermediate car between itself and an intersection regulated by stop signs, the following parameters are used by IDM-IM

$$\begin{cases} s = \sigma - S \\ \Delta v = v \end{cases}$$

where σ is the current distance to the intersection and S is a safety margin, accounting for the gap between the center of the intersection and the point the car would actually stop at. Thus, compared to the IDM, the distance from preceding vehicle is substituted by the distance to the point the vehicle has to stop at. On the other hand, the speed difference is set to the current speed of the car v, so that the stop sign is seen as a still obstacle. This allows vehicles to freely accelerate when far from the next intersection, and then to smoothly decelerate as they approach a stop sign. Once a car is halted at a stop sign, it is informed by the macroscopic level description of the number of cars already waiting to cross the intersection from any of the incoming roads. If there are no other cars, the vehicle may pass. Otherwise, it has to wait for its turn in a first-arrived-first-passed and right hand rule policy.

When a vehicle is heading towards a traffic light intersection, it is informed by the macroscopic description about the state of the semaphore. If the color is green, passage is granted and the car maintains its current speed through the intersection. If the color is red, crossing is denied and the car is forced to decelerate and stop at the road junction, using the modified IDM parameters as in the case for a stop sign.

It may also be stressed out that vehicles behavior can dynamically vary in presence of traffic lights, according to red-to-green and green-to-red switches. In the former case, a car currently decelerating to stop at a red light will accelerate again if the semaphore

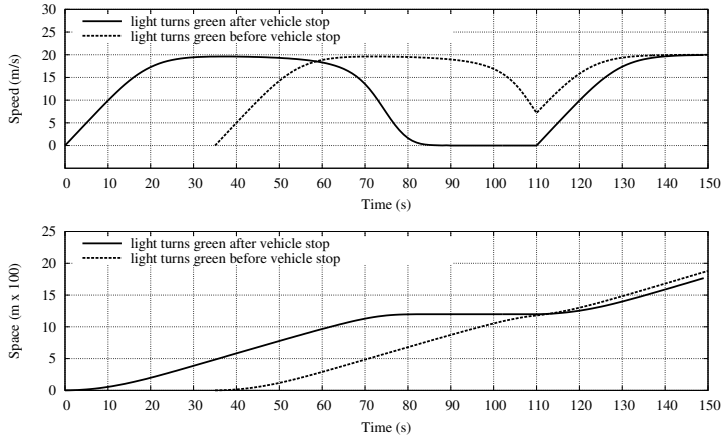


Fig. V-6. Traffic light *red-to-green* scenario. A vehicle, driven by the IDM-IM setup in Table V-1, starts its movement from zero speed, and travels towards a red traffic light. The upper figure shows the evolution of speed in time, while the lower one depicts the car movement on the road versus time.

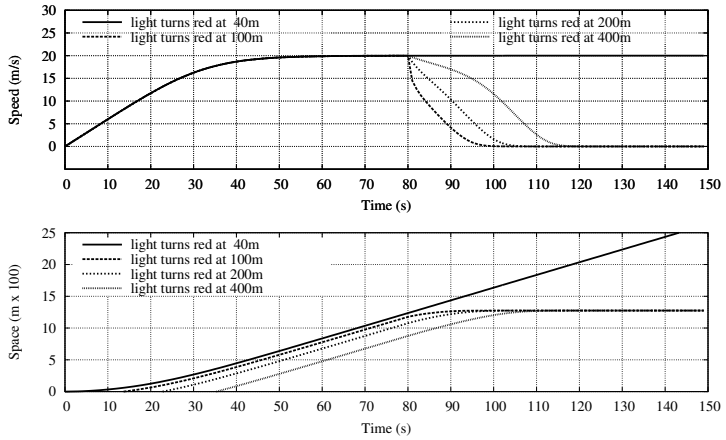


Fig. V-7. Traffic light *green-to-red* scenario. A vehicle, driven by the IDM-IM setup in Table V-1, starts its movement from zero speed, and travels towards a green traffic light, which turns into red at time $t = 80\,s$. The upper figure shows the evolution of speed in time, while the lower one depicts the car movement on the road versus time.

turns green before it has completely halted (see Fig. V-6). In the latter case, a vehicle keeping its pace towards a green light will try to stop if the light becomes red before it has passed through the intersection. In this last case, a minimum breaking distance $\bar{s}$ is evaluated by means of simple kinematic formulae as

$$\bar{s} = v\,t - \frac{\kappa b}{2}\,t^2 = v\left(\frac{v}{\kappa b}\right) - \frac{\kappa b}{2}\left(\frac{v}{\kappa b}\right)^2 = \frac{v^2}{2\kappa b}$$

which describes the space needed to come to a full stop as a function of the current speed of the vehicle, v, the time t and the deceleration value, κb. The last parameter represents the maximum safe deceleration, i.e., the IDM comfortable braking value b scaled by a factor $\kappa \geq 1$. The final expression above is obtained by substitution of t with $(v/\kappa b)$, which is the time at which a zero velocity is reached by inducing a constant deceleration κb on current speed v. Upon computation of $\bar{s}$, if the vehicle finds that it is not possible to stop before the intersection, even braking as hard as possible, i.e., if $\bar{s} > \sigma - S$, then it crosses the intersection at its current speed. Otherwise, it stops by applying a strong enough deceleration. This reproduces a real world situation, since, when a traffic light switches to red, drivers only stop if safety braking conditions can be respected. Examples of driving behaviors in presence of a green-to-red semaphore are shown in Fig. V-7.

The second model we introduce is named *Intelligent Driver Model with Lane Changes* (IDM-LC), and extends the IDM-IM model with the possibility for vehicles to change lane and overtake each others, taking advantage of the multi-lane capability of the macro-mobility description detailed in Section . Two issues are raised by the introduction of multiple lanes: the first is the separation of traffic flows on different lanes of the same road, while the second is the overtakings model itself.

As far as the first problem is concerned, vehicular flows on parallel lanes of the same road are separated by forcing the car following model to only consider vehicles traveling on the same lane. However, as the number of lanes can vary from one road to another, a vehicle approaching a crossroad will receive from the macro-mobility description the information about the structure of the road it is going to move to. It can then adopt one of the following behaviors:

- if the lane the vehicle is currently moving on is also present in the next road on its path, then it moves through the intersection and keeps traveling on the same lane in the next street;
- if the lane currently used by the vehicle does not exist in the next road, then it tries to merge to its right as it approaches the junction. If it cannot do it, e.g. because the lane to its right is very crowded, it stops at the intersection and waits until a spot becomes available.

On the overtaking model itself, the MOBIL model [101] is employed, mainly due to its implicit compatibility with the IDM. This model adopts a game theoretical approach to address the lane changing problem, allowing a vehicle to move to a different lane if the lane change minimizes the vehicles overall braking. Such requirement is fulfilled when the two conditions

$$a^l - a \pm a_{bias} > p\left(a_{cur} + a_{new} - a^l_{cur} - a^l_{new}\right) + a_{thr} \qquad \text{and} \qquad a^l_{new} > -a_{safe}$$

are verified. In the left hand inequality, a is the current acceleration of the vehicle, i.e., $\frac{dx}{dt}$ in the IDM formulae, while a^l is the equivalent acceleration, computed in the case the vehicle moved to an adjacent lane l. Similarly, a_{curr} and a^l_{curr} describe the acceleration of the car which currently follows the vehicle we are considering in the case the vehicle under study stays on its lane, or in the case it moves on another lane l. Finally, a_{new} and a^l_{new} represent the acceleration of the car which would become the new back vehicle if the car under study changed its lane to l, before and after a possible lane change of the latter. The model allows a vehicle to move to lane l if the left hand inequality is verified, that is, if, in terms of acceleration, the advantage of the driver who changes its lane $a^l - a$, is greater than the disadvantages of the following cars $a_{cur} - a^l_{cur}$ and $a_{new} - a^l_{new}$. The MOBIL model also consider a politeness factor p, which scales the right hand term, in a way that, for values of p towards (or above) one, a polite behavior towards other drivers is maintained, while, as p moves to (or below) zero, the driver can become selfish or even malicious. The threshold acceleration a_{thr} introduces a minimum acceleration advantage to allow a lane change, in order to avoid lane hopping in border cases. The bias term a_{bias} is instead added to favor movements to one side: in our case, this bias value is added to the advantage computed for movements to the right and subtracted for movements to the left, thus reproducing the real world tendency of drivers to stay on their right on a multi-lane road. Finally, in any case, the safety condition expressed by the right hand side equation above must be verified for the lane change to occur, meaning that the new back vehicle does not have to brake too hard (its deceleration must be over the safe value a_{safe}) as a consequence of the lane change.

F VANETMOBISIM VALIDATION

Several tests were run on the vehicular movement traces produced by CanuMobiSim and VanetMobiSim, in order to verify that the overall mobility description provided by these tools is able to model vehicular traffic with a sufficient level of realism. This also gives us the possibility to comment on the different outputs obtained with various microscopic mobility models implemented by CanuMobiSim and by VanetMobiSim.

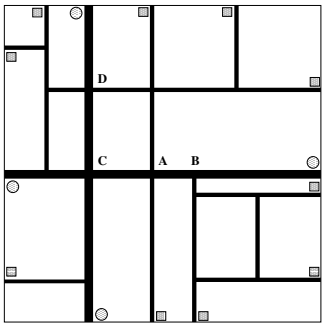


Fig. V-8. City section topology

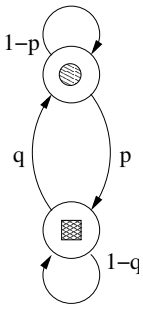


Fig. V-9. Activity chain

First, different micro-mobility models are tested on a user-defined graph representing a square city section of 1500 m side. The urban topology employed in those tests is shown in Fig. V-8, where, unless specified differently, all roads have a single lane,

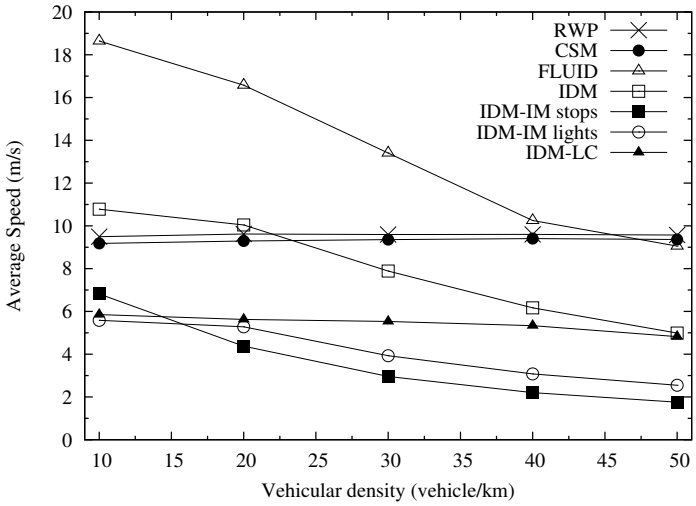


Fig. V-10. Average speed versus vehicular density

and a speed limit of 15 m/s (54 km/h), except for the roads represented with thicker lines, which allow a maximum speed of 20 m/s (72 km/h). Vehicles travel between entry/exit points at borders, identified with circles and squares, crossing the city section according to the fastest path to their destination. The trips generation scheme is activity-based (see Section), and the relative transition probability matrix describes a simple activity chain, depicted in Fig. V-9. There, the states denote the class of the selected destination: a round for the entry/exit points of high-speed roads, a square for the entry/exit points of normal-speed roads, as also shown in Fig. V-9. The chain is trivially ergodic, with steady state $\left(\frac{p}{p+q}, \frac{q}{p+q}\right)$. In our simulation, the probabilities are set so that $p = q = 1/2$, resulting in a stationary distribution $\left(\frac{1}{2}, \frac{1}{2}\right)$. This, along with the proportion between the number of entry/exit points of the two classes, determines a popularity of high-speed roads entry/exit points which is more than double with respect to that of normal-speed entry/exit points. This mimic the tendency of traffic flows to concentrate on the main, high-speed roads. The number of cars traveling at the same time within the city section ranges from 100 to 500, reproducing light (10 vehicles/km) to heavy (50 vehicles/km) traffic conditions. For each test, a single simulation was run, with statistics recorded for 3600 s, after a transient phase of 900 s. When computing 95% confidence intervals for mean values collected averaging in time and on the whole road topology, the error margin was found to be within 0.5% from the mean. The mobility models parameters used in these experiments are listed in Table V-1.

In the rest of this section, we first compare vehicular mobility patterns generated by VanetMobiSim with those of other popular vehicular models, then we validate VanetMobiSim against TSIS-CORSIM, a benchmark traffic simulator.

Model	RWP		CSM	
Parameter	speed	pause	speed	pause
Value	$unif[10,20]m/s$	$unif[0,60]s$	$unif[10,20]m/s$	$unif[0,45]s$

(a) RWP and CSM

Model	IDM				
Parameter	v_0	s_0	T	a	b
Value	$unif[10,20]m/s$	$1m$	$0.5s$	$0.6m/s^2$	$0.9m/s^2$

(b) IDM

Model	FTM			IDM-IM	IDM-LC		
Parameter	s_{min}	s_{max}	κ_{jam}	κ	a_{bias}	p	a_{thr}
Value	$3m/s$	$20m/s$	$0.125car/m$	5	$0.2m/s^2$	0.5	$0.2m/s^2$

(c) FTM, IDM-IM and IDM-LC

TABLE V-1. PARAMETERS USED FOR THE MICRO-MOBILITY MODELS

F.1 Validation against Popular Vehicular Models

In this section, we validate VanetMobiSim by showing how it is able to produce mobility patterns more realistic than those produced by other popular vehicular mobility models. In the following, we also report results obtained with the Random Waypoint Model (RWP), in order to provide a benchmark of this popular model. Due to its nature, this model is not bound by road constraints.

In Fig. V-10, the trend of the average speed versus the number of vehicles is shown. RWP and CSM, ignoring car-to-car interactions, are not affected by the number of vehicles present on the topology, leading to an unrealistically constant mean speed. The mean velocity recorded with CSM is slightly lower than that measured with RWP, even if the mean pause time is shorter in CSM than in RWP. The reason is that CSM limits nodes movement to the road topology, with pauses at every intersection encountered on the path. Thus, the average distance between subsequent pauses is reduced in CSM, at the consequence of a lower average speed.

The low level of realism of these models is further evidenced in Fig. V-11 and Fig. V-12, depicting the time-averaged vehicular density distributions over the road topology obtained with RWP and CSM, respectively. These distribution plots, as well as the equivalent ones for the other mobility models in the remainder of this Section, refer to the 30 vehicle/km case.

As expected, RWP spreads nodes all over the square area, with a higher density of nodes in the center of the map, which is part of RWP normal behavior [128].

On the other hand, cars driven by CSM follow the road topology, and we can observe a non-zero density only where roads are present. Also, in Fig. V-12 the effect of the activity-based mobility can be observed: the two faster and more frequented roads ex-

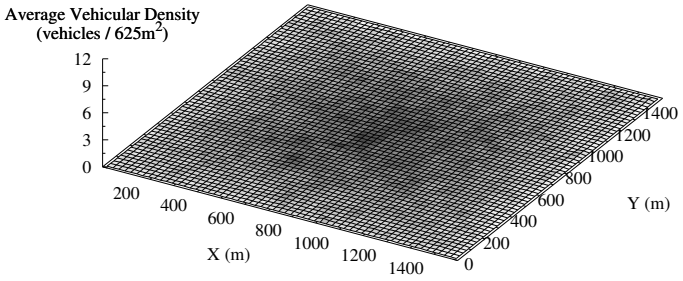


Fig. V-11. Vehicular density: RWP

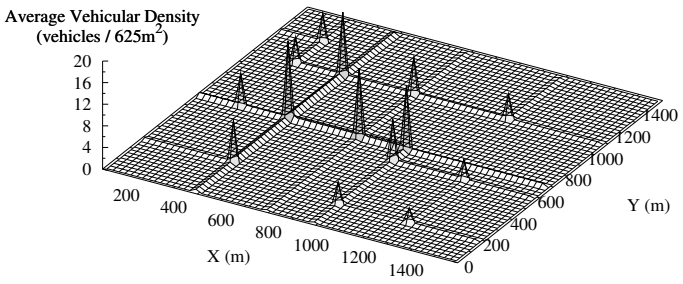


Fig. V-12. Vehicular density: CSM

perience a higher vehicular density with respect to the other streets in the topology. The same can be observed also in the vehicular density plots obtained with the other micro-mobility models. However, CSM produces what we call an *on-off* behavior, with a constant vehicular density on roads and sudden high peaks (note the different density scale with respect to the equivalent plots of the other micro-mobility models) at intersections, where vehicles overlap and stop for a random amount of time. The absence of car-to-car interaction leads thus to an unrealistic complete absence of queuing or acceleration/deceleration phenomena in proximity of intersections.

Looking back at Fig. V-10, modeling the vehicular mobility with FTM produces a very high average speed, mostly due to the fact that vehicles never stop with this model, as the zero speed condition would cause a deadlock as discussed in Section E.2. Probably, a smaller value of the κ_{jam} parameter would have reduced this effect, producing a lower and more realistic figure of the average velocity. However, the settings we chose force vehicles to move at a minimum speed of 10 km/h when they are at a distance of 3 m or less from each others, which represents a suitable real world condition. As expected, FTM reproduces the average speed reduction caused by the vehicular density growth, since the increase of the number of cars traveling concurrently on the same road reduces the fluid speed. However, the vehicular density distribution depicted in Fig. V-13 demonstrates the non sufficient realism of this model. In the considered scenario, a high density is experienced by the central segment marked as AB in Fig. V-8, which is shared by many of the possible paths drivers can choose from. The high quantity of cars driving through determines a reduction of the speed according to the model and creates an even higher vehicular density, which is consistent with what would happen

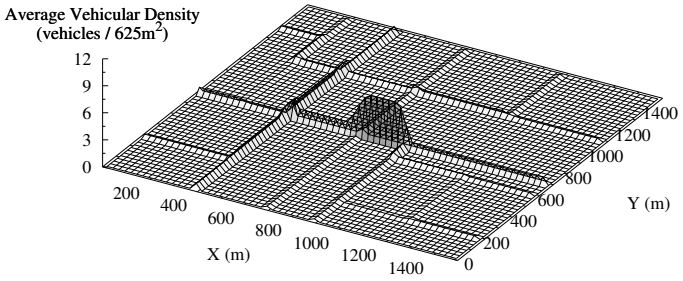


Fig. V-13. Vehicular density: FTM

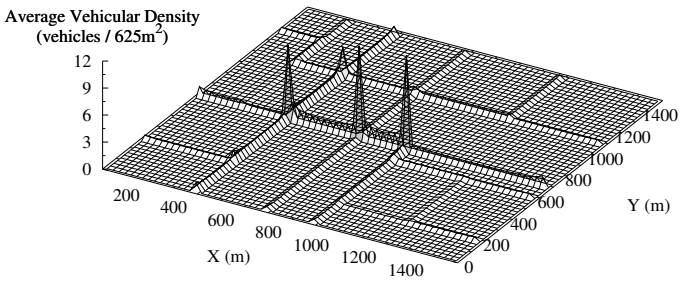


Fig. V-14. Vehicular density: IDM

in a real world situation. However, FTM reasons on a per-edge basis and produces a constant car density over each street, which results in the absence of traffic correlation over connected roads. In our case, it can be noticed that the high car density in AB suddenly disappear in roads after intersections A and B (see Fig. V-8 for the mapping of letters to intersections). Moreover, as FTM ignores intersections, the average number of vehicles at crossroads does not differ from that of vehicles on roads nearby, which again, is far from reality.

As far as IDM is concerned, the average speed curve in Fig. V-10 shows lower values when compared with that obtained with FTM, and quite surprisingly, appears to be affected by the number of cars present on the topology. The speed reduction with respect to FTM is imputable to a more realistic car-to-car interaction, which leads to queuing of fast vehicles behind slow cars. The dependence from vehicular density has instead a two-fold nature: first, the higher density increases the probability of encountering slow vehicles, which generate queues and force a reduction on other drivers' speed. Second, there exists a side effect of the CanuMobiSim implementation, that occurs when vehicles coming from different directions and overlapping at intersections suddenly notice that the safety distance condition is violated. According to the current implementation, they stop and wait for a distance s_0 to be restored before leaving the junction. Such a circumstance causes the average speed to decrease, and occurs more and more frequently as the vehicular density grows. In Fig. V-14, the vehicular density proves that the realism of an accurate car-to-car interaction model in urban scenarios is low, if intersection management is not taken into account. Spikes at highly frequented intersections A, B and C are to impute to the implementation issue explained above,

while in general we can state that IDM does not perform more realistically than FTM in an urban context.

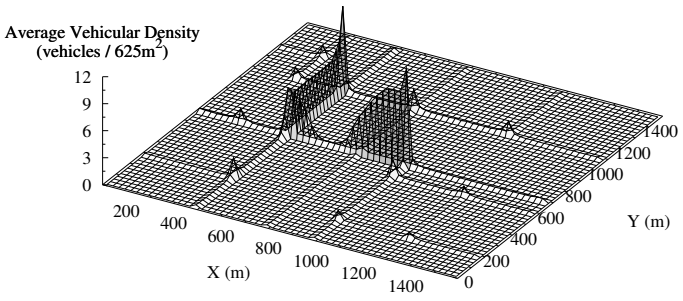


Fig. V-15. Vehicular density: IDM-IM stops

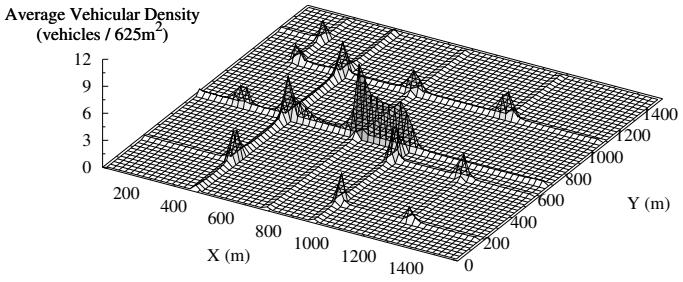


Fig. V-16. Vehicular density: IDM-IM lights

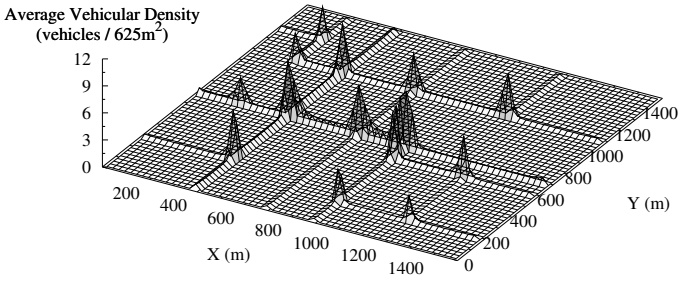


Fig. V-17. Vehicular density: IDM-LC lights

Two different tests were run for IDM-IM, the first with intersections regulated by stop signs, and the second with traffic lights at road junctions. As observed in Fig. V-10, in the first case the model produces a very low average speed, since cars spend most of their time queued at intersections. The problem is exacerbated as the density of vehicles increases and causes longer queues. This can also be noticed by looking at the vehicular density in Fig. V-15, where high vehicular densities, accounting for long queues, are recorded in the neighborhoods of the main intersections A, B, C and D. The higher concentration of vehicles around these intersections also has the side-effect of reducing the number of vehicles on the other roads of the topology, which, as a

matter of fact, record lower vehicular densities. A realistic effect of smooth vehicular density, increasing towards the congested crossroads, is obtained with this model. It can be noticed that such effect in not limited to single segments as it happened with FTM, but also impacts adjacent roads.

When traffic lights with a period of 90 s are used to regulate traffic at intersections, as proved by Fig. V-10, vehicular mobility is improved with respect to the stop sign case, especially in dense scenarios. This could be expected, as traffic lights replace the slow "taking-turns" crossroads management induced by stop signs with a faster "burst" mechanism, in which groups of cars are allowed to cross the junction one after the other, thus saving on acceleration delay. However, for the same reason observed in the stop sign case, the mean speed is still reduced when more cars are introduced in the road topology. An interesting effect can be observed when the vehicular density is low: the stop sign case outperforms the traffic light one. This occurs because, when the number of cars is small, the probability that a crossroad is free is high, thus passage is often immediately granted with a stop handling of intersections, yet at the cost of slowing down and accelerating again. On the other hand, when a traffic light management is considered, vehicles still have to stop in presence of red traffic lights and wait for the light to turn green, even if there are no other cars waiting to cross the intersection. The vehicular density, presented in Fig. V-16, appears consistent with the speed figure, as queuing at highly visited intersections is still present, but noticeably reduced with respect to the previous IDM-IM scenario. Thanks to the improved distribution of traffic over the whole topology, the queuing phenomenon can now be observed at minor intersections, where vehicles have to wait for green traffic lights.

Finally, we report the results obtained when IDM-LC is employed as micro-mobility model. We considered two per-direction lanes on each road, and traffic lights at intersections. From Fig. V-10, modeling vehicular micro-mobility with IDM-LC seems to avoid most of the speed decay effects previously discussed. This is an interesting result, motivated by the fact that i) vehicles actually employ overtakings to avoid slow cars and congested lanes, thus increasing the average velocity, and ii) the presence of multiple lanes helps vehicular mobility in presence of densely populated intersections, as multiple cars can pass through the intersection at the same time and reduce the bottleneck effect of road junctions. In other words, the availability of two parallel unidirectional lanes on each road does not only physically double the capacity of the urban infrastructure, leading to a halved perceived vehicular density, but also brings important correlated effects. In our case, the maximum simulated density of 50 vehicles/km would appear, for the reasons explained before, as a density of less than 25 vehicles/km, a condition which does not seem to generate severe traffic congestion. The vehicular density measured with IDM-LC is depicted in Fig. V-17 and shows that queuing phenomena at intersections are almost equally distributed over the whole topology. Minor intersections experience a higher density with respect to the IDM-IM case as, in absence of critical congestion situations at main junctions, vehicles are more uniformly spread and their presence at smaller crossroads is more noteworthy.

In a different test, we exploited the vehicular mobility description provided by VanetMobiSim to recreate a typical effect of vehicular traffic. In Fig. V-18, the shock waves produced on vehicular density by a periodic perturbation are shown. This result has been obtained with IDM-LC on a 1 km long, unidirectional, double lane, straight road. Cars move towards positive abscissae and a traffic light, located halfway and with a

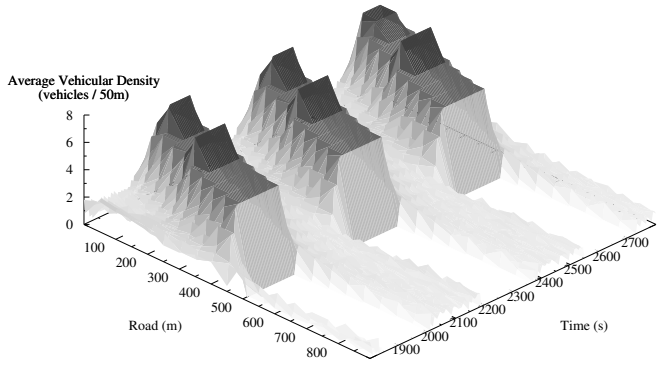


Fig. V-18. Vehicular density shock waves

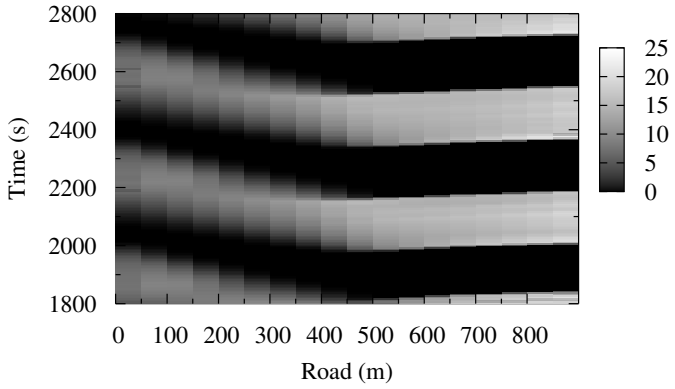


Fig. V-19. Vehicular speed shock waves

period of 360 s, is used as the perturbation source. We can notice that the red traffic light inhibits the movement of vehicles, causing them to stop at 500 m. As more vehicles approach the traffic light, a queue is formed, as shown by the increasing vehicular density, but, when the traffic light turns green, queued vehicles start flowing towards and through the second half of the road. It is possible to see that the high density shock wave propagates in the opposite direction with respect to movement of cars as time goes on. The speed dynamics recorded during the same experiment are depicted in Fig. V-19, where we can observe the queuing perturbation even better, which are represented by the dark, zero-speed areas, propagating against the traffic flow direction in time. Shock waves are a common phenomena of real world traffic. When long queues form in proximity of perturbation sources (crowded intersections, toll stations, in-flow ramps, etc.) the finite reaction time of drivers determines a delay in the propagation of movement. Thus, vehicles queued far from the perturbation origin experience changes in velocity or local traffic density only a long time after the original mobility change occurs at the perturbation.

F.2 Validation against a Benchmark: TSIS-CORSIM

In the previous section, we illustrated how VanetMobiSim was able to reproduce more realistic mobility patterns than other widely spread models used by the community. However, we may raise the question of what *realistic* means. One solution in order to verify the realism of a mobility model is to compare its synthetic traces with real mobility traces. However, those real traces are hard to obtain. Another solution is to compare the synthetic traces with a traffic simulator, which has already been calibrated and validated based on real traces. In the following section, we therefore use TSIS-CORSIM, a benchmark traffic generator within the traffic engineering community, in order to validate the traces generated by VanetMobiSim.

TSIS-CORSIM [75] is a comprehensive traffic simulator, applicable to surface streets, freeways, and integrated networks with a complete selection of control devices (i.e., stop/yield sign, traffic signals, and ramp metering). It simulates traffic and traffic control systems using commonly accepted vehicle and driver behavior models. CORSIM has been validated by showing its ability to model identical mobility patterns to real traces gathered in predefined testing areas. CORSIM has been applied by thousands of practitioners and researchers worldwide over the past 30 years and embodies a wealth of experience and maturity. Funded by the US Federal Highway Administration (FHWA) throughout the last three decades, TSIS-CORSIM has evolved into a benchmark within the transportation profession. We validated VanetMobSim against CORSIM in the version 5.1.

CORSIM has been created for transportation, traffic, and civil engineers. As no interaction has been created in CORSIM for network analysis or simulations, we had to create a specific parser to extract vehicular mobility information. Formally, CORSIM does not output any other data than statistics. However, it communicates with the visualization tool TRAFVU using a set of files from where we extracted the mobility traces. We configured CORSIM according to the same urban topology and activity chain as in Fig. V-8 and Fig. V-9. As CORSIM has been designed to model urban traffic in a high level of precision, it also contains a large set of configuration parameters. For parameters common to VanetMobiSim and CORSIM, we used the same values as in Table V-1.

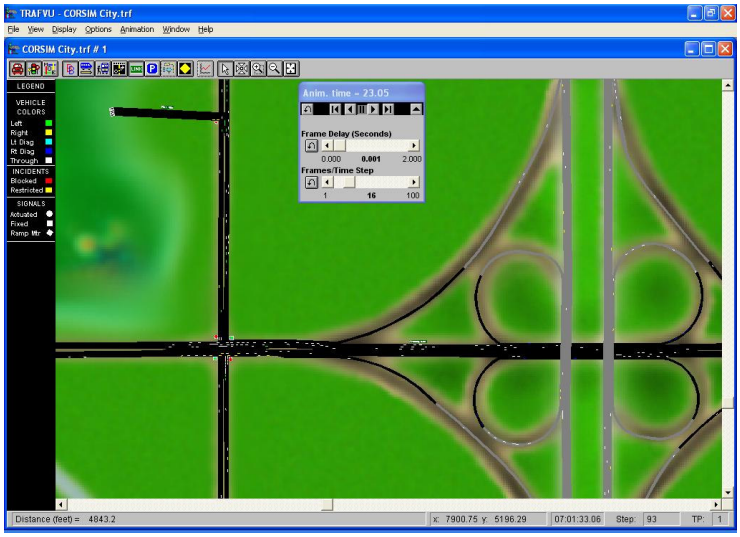


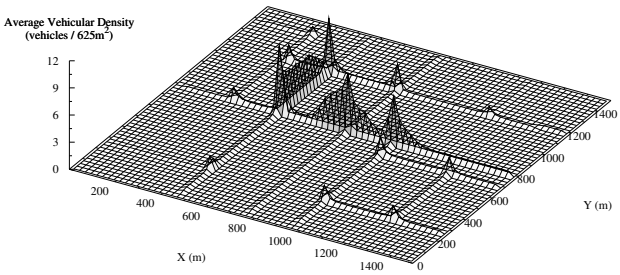
Fig. V-20. CORSIM Vizualizer

For the other parameters, we kept the values defined by default in CORSIM. The exact number of cars simulated by CORSIM cannot be easily configured. Accordingly, we cannot guarantee that we have the same number of cars in both cases. That will be visually seen in the next figures as the local density will slightly differ. However, we are more interested in the geographical distribution of the cars than in local intensity.

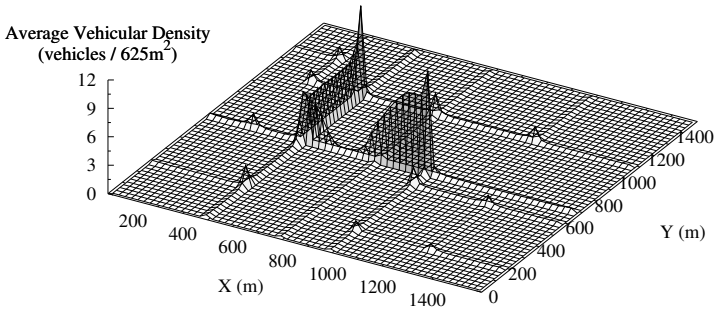
In Fig. V-21, we compare the spatial distribution of vehicles in the topology generated by CORSIM and by VanetMobiSim. Even though the local densities are not similar, we can clearly see that the aggregation occurs at similar places. Similarly to VanetMobiSim, vehicles are likely to follow streets with a higher velocity. Accordingly, we see that in both figures, a bottleneck is generated on the *CD* and *AB* edges. Indeed, the intersection at *C* is a major crossroad between two high speed streets, and the intersection at *B* contains a side street which potentially attracts a lot of traffic due to the number of *attraction points*. As all intersections are modeled by stop signs where cars pass one at a time according to the right hand rule, the slow traffic flow at those intersections are generating similar mobility patterns.

Figure V-22 compares the spatial distributions between CORSIM and VanetMobiSim when intersections are controlled by traffic lights. Similarly to the previous graph, we can clearly see that the spatial distribution is very similar, cars aggregating in the same intersections or road-segments. We can also observe a similar effect of the traffic light located at the intersection *C*, which helps resolving the vehicle aggregation on the edge *CD* both in CORSIM and VanetMobiSim.

Finally, in Fig. V-23, we see that the added lane changing capability has a similar effect on traffic aggregation. In both cases, the large aggregations are reduced to local peak densities at similar intersections. The local densities are also more uniformly distributed per intersection. By allowing cars to overtake slow and potentially blocking cars, both CORSIM and VanetMobiSim manage to reduce the clustering effect at the intersection very similarly. Accordingly, even though the two simulators use different

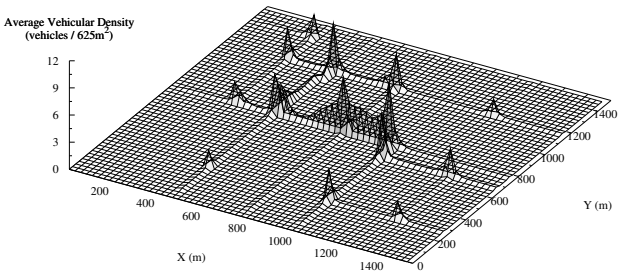


(a) CORSIM

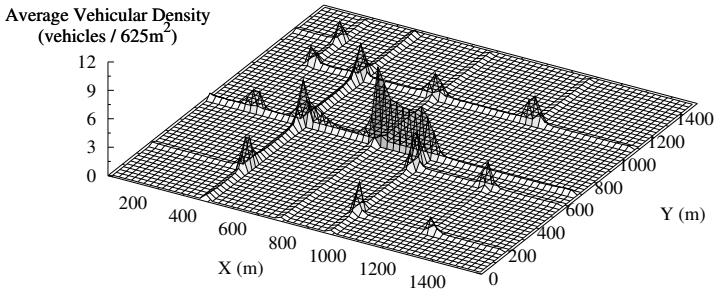


(b) VanetMobiSim

Fig. V-21. Comparison of the spatial distribution with Stop signs

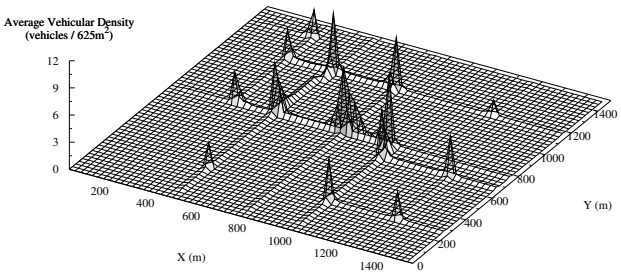


(a) CORSIM

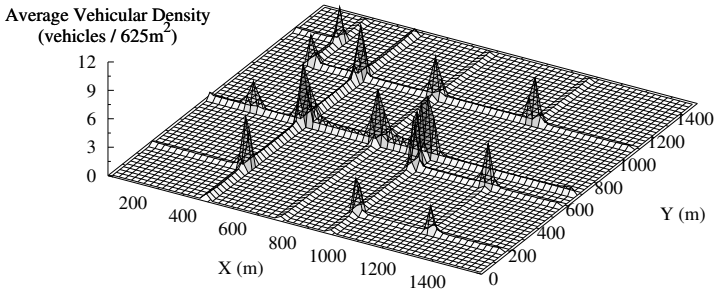


(b) VanetMobiSim

Fig. V-22. Comparison of the spatial distribution with Traffic lights



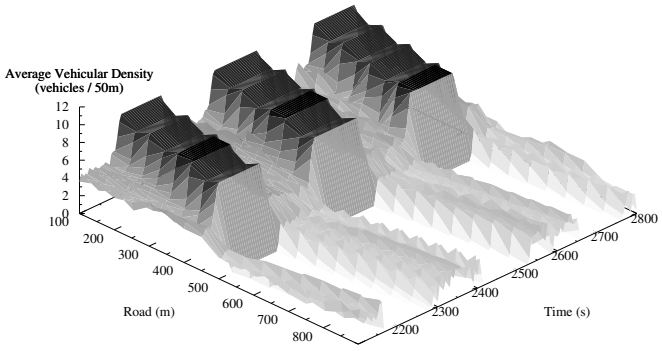
(a) CORSIM



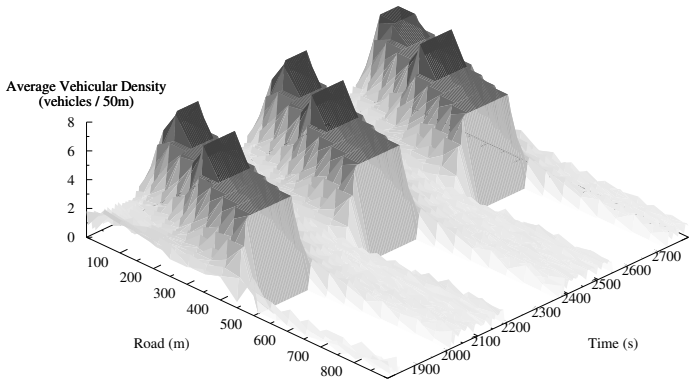
(b) VanetMobiSim

Fig. V-23. Comparison of the spatial distribution with Traffic lights and lane changing

macro- and micro-mobility models, potentially having different configuration parameters, we see that CORSIM and VanetMobiSim produces similar traffic distributions.



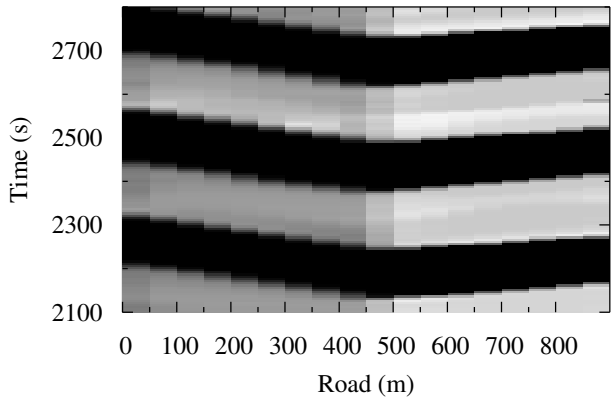
(a) CORSIM



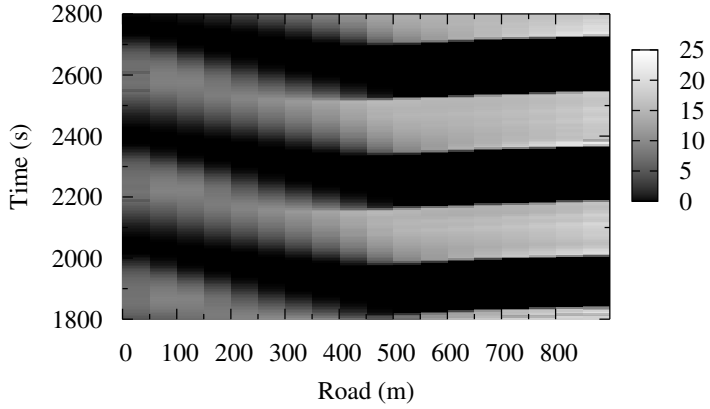
(b) VanetMobiSim

Fig. V-24. Comparison of the density shock waves

In the next set of figures, we compare the shock wave effect created by a periodic perturbation modeled by a traffic light. The topology is identical to the one used for the Fig. V-18 and Fig. V-19. By comparing the two plots in Fig. V-24, we first can see that CORSIM and VanetMobiSim generate similar density shock waves. In both cases, the local perturbation is backward propagated. In Fig. V-25 the similarity is even more exacerbated as we see the periodic speed perturbation. In both plots, the speed shock waves are also backward propagated according to the delayed stop and move patterns observable in waiting queues. Similarly to the traffic distribution previously displayed, CORSIM and VanetMobiSim produces very similar mobility patterns generated by periodic perturbations.



(a) CORSIM



(b) VanetMobiSim

Fig. V-25. Comparison of the speed shock waves

To conclude this section, we would like to emphasize that although CORSIM and VanetMobiSim neither use similar micro-mobility patterns, nor are controlled by same configuration parameters, we showed that the traffic distribution and well as the shock waves generated by a periodic perturbation were similar and conformed to real life situation. Accordingly, this let us claim that the mobility patterns generated by VanetMobiSim are validated and realistically reflect real motion patterns.

F.3 Illustration in Real Urban Case

As a further addition to the validation of the mobility generated by VanetMobiSim, Fig. V-26 shows a snapshot of the vehicular mobility obtained with VanetMobiSim on the urban area of Westwood in Los Angeles overlapped with a real map of the same city section. The simulated vehicular mobility is extracted from the *nam* network animator of *ns-2*. The snapshot refers to a simulation involving IDM-IM, traffic lights at intersections, a random speed-based path selection. Although TIGER maps do not include speed limits information, we deduced them from the street class, according to the local speed limitation policy. Drivers thus take into account the path length and the allowed speed along the path, making detours if a path appears globally faster. The consequence can be seen in in Fig. V-26, where Wilshire Boulevard attracts the majority of drivers, hoping to save time by using a large East-West commuting corridor instead of parallel streets. When the local vehicular density exceeds the traffic lights management capability, like at intersections between Wilshire and Glendon, and Glendon and Lindbrook, the traffic cluster pours out and cars start stacking up on the surrounding streets and not only at the road junctions. These congestion phenomena can be easily observed in real-life situations.

Fig. V-26. Simulated vehicular mobility in the Westwood area

G CONCLUSION

As the lack of realism of the Random Waypoint Model (RWM) was a serious obstacle to the outcome of the predictability analysis of Chapter IV, we decided to assess more

realistic mobility patterns. Vehicular motions seemed an appropriate choice as their patterns were complex enough and reflecting real situations, but also showed some signs of regularity which could be exploited by mobility prediction heuristics.

In this Chapter, we therefore presented VanetMobiSim, an open extension to the CanuMobiSim user mobility framework capable of producing realistic vehicular mobility traces for several network simulators. We reviewed the macroscopic and microscopic mobility descriptions of CanuMobiSim, and detailed the additions to both scopes brought by VanetMobiSim. Simulation results were presented and discussed, trying to understand the differences between various micro-mobility models, in terms of vehicular density and speed distribution.

By taking a comprehensive look at the results obtained, it appears clear that the detail level of the micro-mobility models implemented by other mobility models available for VANETs is not sufficient to reproduce realistic vehicular traffic traces. The increased degree of detail introduced by the micro-mobility models of the VanetMobiSim extension, and the possibility of their interaction with the new macro-mobility description appear necessary to reproduce real world phenomena. In particular, the progressive introduction of stops signs, traffic lights, multiple lanes and overtakings demonstrates how the modeling of each of these features brings noticeable changes to the system performance.

Moreover, we compared the vehicular mobility traces obtained by VanetMobiSim with TSIS-CORIM, a benchmark traffic generator able to reproduce realistic and validated vehicular traffic traces. Through the illustration of the similarity between both traces, it let us claim that VanetMobiSim is able to produce realistic vehicular mobility traces available to network simulators. That makes VanetMobiSim one of the few synthetic vehicular-oriented mobility simulator fully validated and freely available to the vehicular networks research community.

From a networking point of view, the differences observed between different micro-mobility models, in terms of vehicles and speed distribution, queuing dynamics and presence and size of clusters may heavily affect the connectivity of VANETs, and, consequently, the performance of ad-hoc network protocols. It is part of our future work to investigate the actual impact of these different traffic phenomena on a vehicular network, so to understand which factors must be considered and which can be neglected for a confident VANETs simulation study.

Also, a very important factor when simulating highly mobile networks is the radio propagation model. Results obtained without accounting for the impact of large obstacles, such as buildings, on the radio signal propagation can hardly be realistic. We are thus interested in studying this aspect, taking benefit from the availability of a detailed topology description to introduce a new component in VanetMobiSim, capable of generating radio propagation information for network simulators.

ACKNOWLEDGMENTS

We would like to thank Prof. Mario Gerla and Dr. Giovanni Pau from the Network Research Lab (NRL) at the University of California in Los Angeles (UCLA) for having granted us the access to TSIS-CORSIM during the Summer 2006.

CHAPTER VI

Analysis of Vehicular Mobility Patterns on Routing Protocols

Contents

Abstract—*In this Chapter, we illustrate how the realistic motion patterns introduced in Chapter V affect the velocity, and how new parameters become necessary to evaluate the performance of routing protocols in Vehicular Ad Hoc Networks (VANETs). To express our point, we evaluate the performance of AODV with realistic urban scenarios. We show how new urban specific*

parameters have significant impacts on routing, and de-facto replace some non-urban specific parameters. For example, the average velocity appears to be irrelevant in urban scenarios and should be replaced by road segment lengths. Then, we evaluate AODV and OLSR performance in realistic urban scenarios. We study those protocols under urban-specific metrics such as road segment length, and cluster effect, or non-urban specific metrics such as vehicle density, and data traffic rates. We show that clustering effects created by cars aggregating at intersections have remarkable impacts on evaluation and performance metrics. We conclude that OLSR is a better candidate than AODV for routing in VANET in urban areas.

Keywords— *Simulation Parameters, Performance Evaluation, Urban Environment, Realistic Vehicular Mobility Models, AODV, OLSR, VANET.*

IN Chapter V, we showed the drastic difference between the motion patterns generated by the VanetMobiSim Model (VMM) and other models used in the literature to evaluate VANET protocols. As the spatio-temporal distribution of the vehicles and speed are key parameters for the stability of wireless links, routing protocols might show different behavior. Accordingly, previous studies of ad hoc routing protocols cannot be applied to VANETs modeled by realistic motion patterns. In this Chapter, we therefore aim at studying the effect of realistic vehicular motion patterns on routing protocols for VANETs.

A MOTIVATION

One of the critical aspects when evaluating routing protocols for VANETs is the employment of mobility models that reflect as closely as possible the real behavior of vehicular traffic. Simple random models cannot describe vehicular mobility in a realistic way, since they ignore the peculiar aspects of vehicular traffic, such as cars acceleration and deceleration in presence of nearby vehicles, queuing at roads intersections or traffic bursts caused by traffic lights. All these situations greatly affect the network performance, since they act on network connectivity, which makes vehicular specific performance evaluations fundamental when studying routing protocols for VANETs. Initial works [172, 173] on performance evaluation were based only on random motions, such as random walk models, and lacked any interaction between cars, generally referred as *micro-mobility*. Following the recent interest in realistic mobility models for VANETs, new studies appeared on performance evaluations of VANETs in urban traffic or highway traffic conditions [174, 118]. As these new models generate urban specific spatial and temporal dependencies, the real mobility parameters differ from the initial and controlled ones. Performance comparison may become unfair and arguable.

Another critical aspect is to use the appropriate parameters in order to evaluate routing protocols. A crucial parameter influencing the performance of Vanets is referred by the generic term *mobility*. In simple models, mobility is equal to velocity. However, on the eve of realistic mobility models, it becomes hard to understand the real parameters controlling this *mobility*. However, only few studies have been done illustrating how realistic motion patterns influence the mobility and other configuration parameters.

B Basic Idea

Our objective is to illustrate how realistic urban motions reduce the effect of some standard evaluation metrics, and how they generate new urban-specific performance parameters never described in the past. Using VanetMobiSim Model (VMM) presented in Chapter V, it becomes possible to evaluate more realistically ad hoc routing performances for vehicular networks. We configure VanetMobiSim to model an urban environment, then evaluate the performance of AODV and OLSR in terms of *(i) Packet Delivery Ratio (PDR) (ii) Delay (iii) Hop Count*. We test AODV and OLSR in four different conditions *(i) velocity (ii) road segment length (iii) cluster effect (iv) traffic load*.

C Summary of Contribution

We first show how the average velocity has a minor impact on performance as it cannot reflect the real velocity in urban traffic. A more significant parameter is the road segment length, as this is the parameter controlling the real velocity. We also exhibit how the clustering effect obtained at intersection has a major effect on the effective average velocity during the simulation. We finally illustrate how OLSR outperforms AODV and is consequently a better candidate than AODV for routing in urban environment.

D Organization of Work

The rest of the Chapter is organized as follows. In Section E, we provide a brief overview of related work in MANET protocol evaluation and comparison. Section F illustrates the effects of VMM mobility patterns on standard performance parameters. In Section G, we evaluate AODV and OLSR performance in realistic urban scenarios, and we finally conclude the Chapter in Section H.

E Related Work on MANET Protocol Comparison

Several studies have been published comparing the performance of routing protocols using different mobility models or performance metrics. One of the first comprehensive studies was done within the framework of the Monarch project [172]. This study compared AODV, DSDV, DSR and TORA and introduced some standard metrics that have been then used in further studies of wireless routing protocols. A paper by Das et al. [173] compared a larger number of protocols. However, link level details and MAC interference are not modeled. Another study [175] compared the same protocols as the work by Broch et al. [172], yet for specific scenarios as the authors understood that random mobility would not correctly model realistic network behaviors, and consequently the performance of the tested protocols. Globally, all these papers concluded that reactive routing protocols perform better than proactive routing protocols.

Although the proactive OLSR protocol has been developed in 2002, very few studies

compared it with other ad hoc network protocols. Clausen *et al.* [176] evaluated AODV, DSR and OLSR in varying network conditions (node mobility, network density) and with varying traffic conditions (TCP, UDP). They showed that unlike previous studies, OLSR performs comparatively to the reactive protocols.

Following the developments started with scenario-based testing, it also became obvious that, as scenarios were able to alter protocol performances, so would realistic node-to-node or node-to-environment correlations. This approach became recently more exciting as VANETs attracted more attention, and a new wave of vehicle-specific models appeared. The most comprehensive studies have been performed within the Fleetnet project [177]. In a first study [174], authors compared AODV, DSR, FSR and TORA on highway scenarios, while [118] compared the same protocols in city traffic scenarios. For instance, they found that AODV and FSR are the two best suited protocols, and that TORA or DSR are completely unsuitable for VANET. Another study [178] compared a position-based routing protocol (LORA) with the two non-position-based protocols AODV and DSR. Their conclusions were that, although AODV and DSR perform almost equally well under vehicular mobility, the location-based routing schema provides excellent performance. Similar results has been reached by members of the NoW project [179], which was their major justification for the design of position-based forwarding techniques. However, to the best of our knowledge, no performance evaluation has been conducted between OLSR and other routing protocols under realistic urban traffic configurations.

F Influence of VanetMobiSim on Vehicular Motion Patterns

The VanetMobility Model (VMM) requires many configuration parameters, all of which have effects on the modeling of vehicular motions. In this section, we illustrate the average *road segment length*, the average *acceleration, resp. deceleration rate*, and the *clustering effect*, which are three major novel motion parameters VMM defines, and compare their influence on the RWM.

With these parameters, VMM generates motion patterns that cannot be modeled by pure random motions. Yet, these parameters deeply influence the spatial distribution and velocity of cars in the network. Indeed, any single one or any combination of them is able to generate a significant difference between the initial average velocity and the real velocity, or between the average and the local density. This problem may be formulated as the difference between initial distribution of the statistics of mobility parameters and the steady state distribution. However, as the problem of analytically computing the steady state distributions of realistic mobility models is much more complex than that of random models, the only way to illustrate this effect is through simulations. The corollary is that any simulation must be undertaken after a sufficiently large "warming" time in order to reduce the effect of the transient state.

F.1 Parameters Definition

Before going further, we would like to define the particular parameters we use in this Chapter.

We first provide *Speed* related definitions

- *Average Speed*– The average speed controls the distribution of the random variable that determines the speed between each destination point.
- *Desired Speed*– The desired speed is the speed sampled at each destination point. It is therefore the speed a driver aims at reaching using a smooth acceleration. However, according to traffic regulations, there is no guarantee that this speed may ever be reached.
- *Real Speed*– The real speed is the temporal speed obtained at each time instant. It is subject to traffic, traffic signs and driver habits.
- *Speed Decay*– The speed decay is the gap between the desired speed and the real speed.

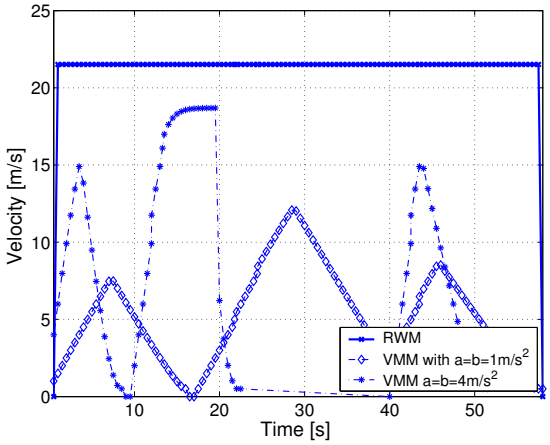
Then, the *Clustering Effect* is a particular parameter specific to realistic mobility models which should not be mistaken with the *density* or the *number of nodes*. Indeed, the clustering effect is a parameter taken from urban traffic modeling and controls the aggregation at the intersections. Our purpose is to spot out the effects solely dependent on the urban traffic distribution and not dependent on effects on the MAC layer or on routing protocols from an increased number of neighbors. Accordingly, the clustering effect is controlled by increasing the number of vehicles in the urban area, while reducing the transmission range in order keep the average network density constant[1] (in terms of average number of neighbors per vehicle). Thanks to it, we are able to see the effect of spatial and temporal dependencies on routing protocols, and not only the effect of the density that has already been studied in the past.

Finally, a *Road Segment* is defined as the piece of road connecting two intersections. The length of a road segment is therefore the distance between two intersections. Its major effect on realistic mobility models is its control of the gap between the desired speed and the real speed. It is also able to control the clustering effect.

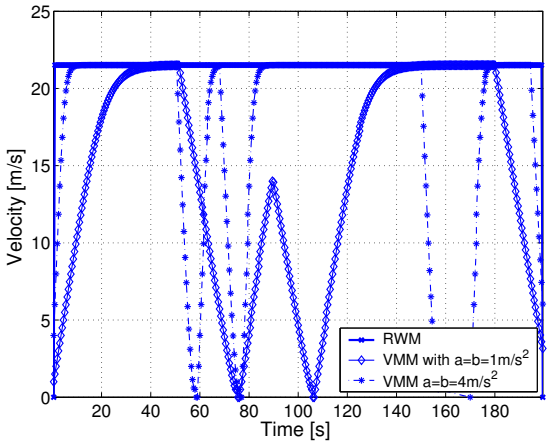
F.2 Illustration

In Fig. VI-1, we illustrate the effects of the average road segment length and the acceleration, resp. deceleration rate, on the real velocities of vehicles. In both figures, the desired velocity is the one reached at any time by RWM, and we modeled the velocity of a single vehicle during on single trip. Unlike the RWM which ignores the VMM's parameters, the velocity modeled by VMM fluctuates significantly as it is influenced by the acceleration rate and the road segment length. By considering the acceleration rate $1m/s^2$ and comparing Fig. VI-1(a) and VI-1(b), vehicles never reach the desired speed

[1] It is possible to obtain a significant performance difference if we have a large clustering effect at a low network density or a low clustering effect at a high network density.



(a) Road segment length=150m



(b) Road segment length=250m

Fig. VI-1. Illustration of vehicular real velocity on a single trip, where a and b are the acceleration, resp. deceleration rate

in the former figure, as cars modeled by VMM respect traffic regulations and must decelerate and stop at each intersection in the trip always before reaching the desired speed. However, by looking at Fig. VI-1(b), the effect may be limited by increasing the distance between two successive intersections, as cars have more time to reach their desired speed. The second parameter is the acceleration, resp. deceleration rate. Considering Fig. VI-1(a), for a fixed distance between two intersections, a car with a strong acceleration rate is quickly going to reach the desired speed and will run faster on the selected road segment than a car with a smaller acceleration rate. Since the real velocity is an important parameter for routing protocols in mobile ad hoc networks, we expect these new parameters to be more fundamental than average, or desired velocities.

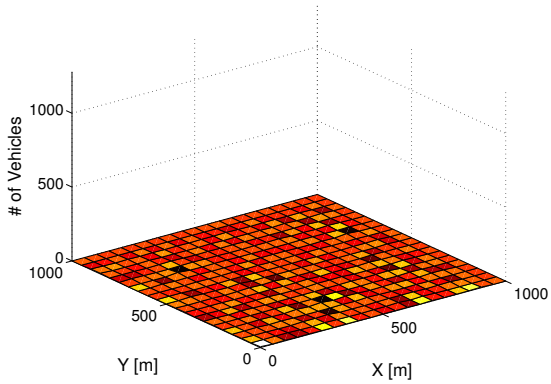
RWM's objective is to keep vehicles position uniformly distributed in the network, an effect that may be sought for SANETs for instance. However, for VANETs, this is seldom the case as vehicles follow predefined paths and aggregate at intersections. This leads to a non-uniform distribution of vehicles in the network, which we call the *clustering effect*. As we see on Fig VI-2(b), the number of vehicles observed in the network is higher on predefined roads and even higher on intersections, while the number of vehicles is, as expected, uniformly distributed in Fig VI-2(a). Since the distribution of vehicles in the network has an impact on connectivity and data dissemination, we also expect the clustering effect to have a significant influence on performance of mobile ad hoc networks in vehicular urban areas.

As an illustration of a possible effect on performance, we show in Fig. VI-3 the average speed decay from a desired velocity that vehicles experience with VMM. However, this desired velocity is subject to speed limitations that cannot be exceeded, or to any obstacle that either reduces the vehicle speed or even forces it to stop. Clearly, the desired speed is always reached by the RWM as there is no correlation with the environment, but VMM needs to comply with those limitations. Accordingly, there is no guarantee that this velocity can even be reached during the simulation under VMM. As we can see on Fig. VI-3(a), there is a drastic decay as a function of the desired velocity, and it does not depend on it. The desired velocity is therefore not able to control the real velocity reached by nodes under VMM. However, as seen in VI-1, the road segment length does. The speed decay is therefore not stable in Fig. VI-3(b), since it is influenced by the road segment length or acceleration, resp. deceleration rates.

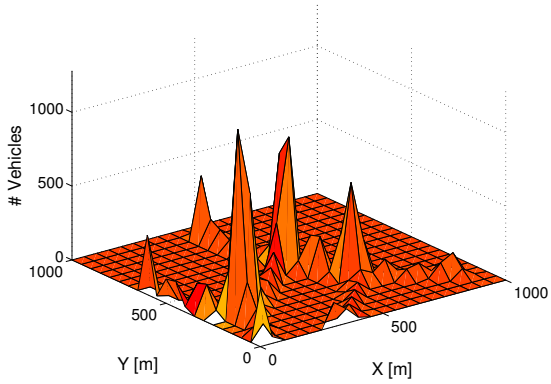
The main conclusion is that network mobility as defined in previous works cannot be used as an evaluation metric for vehicular ad hoc networks. We should rather define new metrics as acceleration/deceleration factors, clustering effect or distance between two intersections.

G PERFORMANCE EVALUATION

In order to illustrate the influence of the new parameters described in the previous section on routing protocols, we used the open source network simulator ns-2 in its version 2.27 as it is widely used for research in mobile ad hoc networks. We first provide a description of the scenarios and then present the obtained results.

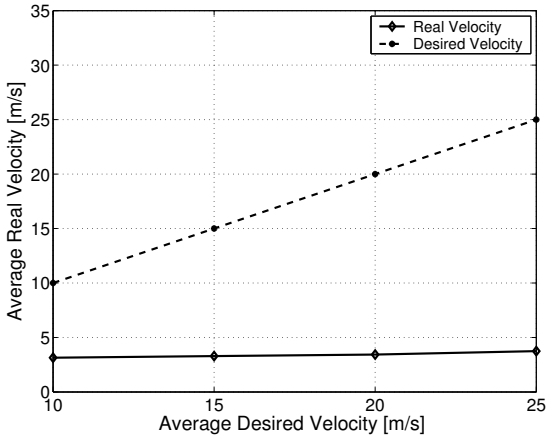


(a) RWM

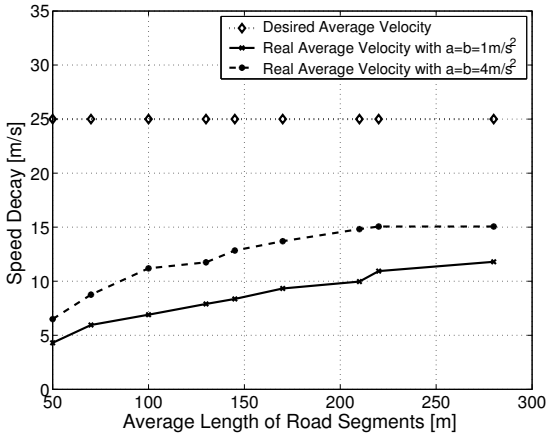


(b) VMM

Fig. VI-2. Spatial distribution of vehicles in the urban environment (Cluster Effect)



(a) Average Desired velocity



(b) Average length of road segments

Fig. VI-3. Illustration of the Speed Decay, where a, resp. b are the acceleration, resp. deceleration rates for RWM (doted curve) and VMM (dashed and plain curves)

G.1 Scenario Characteristics

In this Chapter, we consider squared urban areas of 1000x1000m constituted of three different cluster categories: downtown, residential and suburban. The different obstacle densities for these three categories are summarized in Table VI-2(b). Fig. VI-4 displays an example of an urban graph used in this Chapter. The simulation parameters are given in Table VI-1. We tested each protocol with a spatial model composed of 30% of traffic lights and 70% of stop signs. Finally, each road is composed of 2 lanes in order to let cars overtake if necessary.

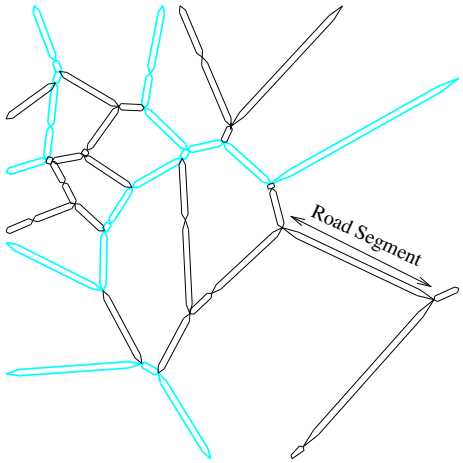


Fig. VI-4. Illustration of an urban graph used for the simulations

Vehicles are randomly positioned on intersections. Then, each vehicle samples a desired speed and a target destination. After that, it computes the shortest path to reach it, taking into account single flow roads. Eventually, the vehicle moves and accelerates to reach a desired velocity according to street regulations. When a car moves near other vehicles, it decelerates to avoid the impact or it tries to overtake them. When it is approaching an intersection, it first acquires the state of the traffic sign. If it is a stop sign or if the light is red, it decelerates and stops. If it is a green traffic light, it slightly reduces its speed and proceeds to the intersection. At target destination, the car decelerates, stops, and then samples a new destination. The different parameters for the micro-model are given in Table VI-2(a).

We decompose our performance analysis into five different scenarios, where parameters are fixed according to Table VI-4. In the first scenario, we want to see the influence of the average velocity. Next, we analyze the effect of different lengths of road segments and also on the clustering effect at intersections. Finally, we study the data traffic rate and the network local density. We use the word "local" density, as it represents the density measured at the intersections and not the average density.

Each point is the average of 10 samples, while the error bars represent a 95% confidence interval. We also point out that in all first three scenarios, we maintain the same average density, as we want to exhibit results not related to an increased density. Finally, for each scenario, we first simulated AODV for the RWM [164] and the VMM with IDM-LC, and then AODV and OLSR but only for VMM with IDM-LC. Accord-

Network Simulator	ns-2.27
Mobility Models	RWM [164], VanetMobiSim (VMM) IDM-LC
AODV Implementation	AODV-UU
$Hello^{aodv}$ Interval	3s
OLSR Implementation	NRLOLSR
$Hello^{olsr}$ Interval	0.5s
TC^{olsr} Interval	2s
Simulation time	1000s
Simulation Area	1000m x 1000m grid
Number of Nodes	$10 \rightarrow 80$
Tx Range	100m
Speed	Uniform
Density	$\#nodes \cdot \frac{\pi \cdot range^2}{X_{dim} \cdot Y_{dim}}$
Data Type	CBR
Data Packet Size	512 bytes
MAC Protocol	IEEE 802.11 DCF
MAC Rate	2 Mbits/s
Confidence Interval	95%

TABLE VI-1. SIMULATION PARAMETERS

Param	Description	Value
a	Maximum Comfortable Acceleration	$0.9m/s^2$
b	Maximum Comfortable Deceleration	$0.5m/s^2$
l	Vehicle Length	5m
s_{com}	Minimum Congestion Distance	2m
t	Safe headway time	1.5s
b_{sav}	Maximum safety deceleration	$4m/s^2$
p	Politeness	0.5
a^{th}	Lane Change Threshold	$0.2m/s^2$
T^{light}	Traffic Light Transition	30s

(a) Micro-model

Clusters	#obstacles per $100m^2$	#cluster per $1000m^2$	cluster ratio
Downtown	50	4	10%
Residential	12.5	4	40%
Suburban	2.5	4	50%

(b) Macro-model

TABLE VI-3. VEHICULAR MOBILITY MODEL PARAMETERS

ingly, we are able to see the effect of realistic urban motions on the parameters and on the performances.

Finally, we would like to emphasize that all simulations have been obtained on sparse networks, as it will be the major case in early stage of VANETs[2], and that an increased density could only further improve the performance of AODV and OLSR. Moreover, this allowed us to illustrate that the particular urban vehicular mobility patterns are able to locally reduce this sparseness and therefore increase the performance of routing protocols.

Before moving to the metric definition, we would like to more precisely describe two scenarios that are novel to performance evaluation in mobile ad hoc networks. Indeed, urban mobility generates particular motion patterns that cannot be accurately illustrated by standard scenarios such as velocity, node density or data traffic. We therefore need urban specific performance evaluation scenarios.

The first scenario is *road segment length* scenario. By increasing the length of road segments from 50m to 300m, we actually model urban traffic distribution observed from small roads in highly urban areas to highways in major commuting corridors. By fixing the average desired velocity and increasing the road length, we increase the time spent by vehicles on the road elements, which in turn reduces the clustering effect and also increases the real vehicular speed. In order to see the sole effect of the length of road segments and not network disconnections, we maintain a fixed node density and increase the transmission range accordingly.

The second scenario is the *cluster effect* scenario. We increase the number of vehicles in the urban area, while reducing the transmission range in order keep the average network density constant (in terms of average number of neighbors per vehicles). We indeed want to spot out results solely dependent on the urban traffic distribution and not on effects on the MAC layer or on routing protocols from an increased number of neighbors. The average road length in this scenario is set to 150m. By increasing the number of vehicles and keeping fixed the average road length, we actually increase the interaction of each car with its environment, which in turn limits its ability to reach a desired speed and accordingly reduces the real speed.

G.2 Metrics Definitions

We measured several metrics for MANETs routing that are mostly influenced my mobility:

- *Packet Delivery Ratio (PDR)*– It is the ratio between the number of packets delivered to the receiver and the number of packets sent by the source.
- *Delay*– It measures the average end-to-end transmission delay by taking into account only the packets correctly received.
- *Hop Count*– It represents the number of hops that a packet has taken before it has been correctly delivered.

[2]We only expect a slow increase of VANET ready vehicles.

Scenarios	Data Rate [Mbits/s]	Network Mobility [m/s]	Nodes Density	Road Length [m]	Nbr. of Nodes	Tx Range [m]
Desired Velocity	0.8	v^{min}=0, v^{max}=20 to v^{min}=15, v^{max}=35	≈ 2	50	60	100
Road Segment Length	0.8	v^{min}=15, v^{max}=35	≈ 2	50 to 280	60	100 to 500
Clustering Effect	0.8	v^{min}=15, v^{max}=35	≈ 2	150	20 to 60	424 to 244
Data Rate	0.02 to 8	v^{min}=15, v^{max}=35	≈ 2	50	60	100
Network Local Density	0.8	v^{min}=15, v^{max}=35	1.96 to 15.7	50	10 to 80	100

TABLE VI-4. SIMULATION SCENARIOS

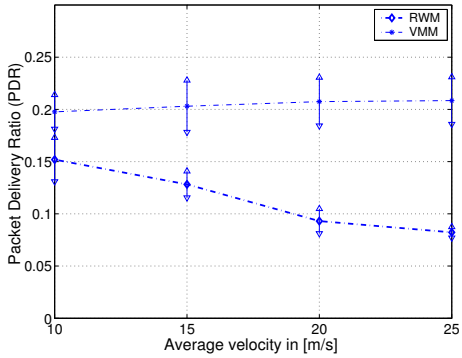
G.3 Influence of Vehicular Mobility Patterns on AODV

In Fig. VI-5(a), we see that for VMM[3], the average velocity does not have any effect on the PDR, which is a surprising result since the velocity is a common metric in performance evaluation, and previous results have shown that AODV was sensitive to it. On the other hand, the performances with RWM are influenced by the velocity and differ significantly from those with VMM. Indeed, we see in Fig. VI-5(b) that an increasing velocity worsens the delay for the RWM, but does not significantly impact the VMM. Similarly, Fig. VI-5(c) illustrates how a higher velocity reduces the number of hops for VMM, but does not conclusively affect RWM.

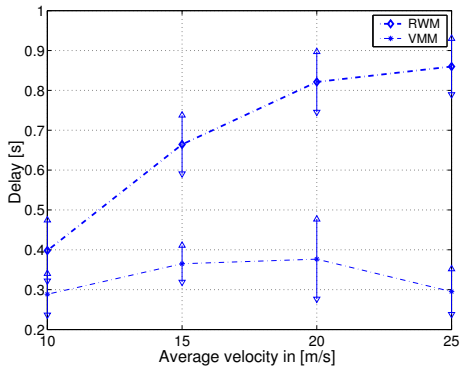
Actually, the explanation for this behavior comes from the micro-model and its interaction with the spatial environment. Indeed, when modeling smooth transitions and realistic interactions with urban traffic regulations, a fixed initial velocity does not make any sense. Instead, we define an *average desired velocity* a driver aims at reaching with a smooth acceleration. However, this desired velocity is subject to speed limitations that cannot be exceeded, or subject to obstacles that reduces vehicle speed or even forces it to stop. Accordingly, there is no guarantee that this velocity can even be reached during the simulation. And, as it can be seen in Fig. VI-3(a), the real speed is stable with respect to the average velocity, and significantly lower than the desired velocity, which explains the relative stability of AODV with VMM.

In the next set of simulations, we illustrate the effect of the average length of road segments on the performance of AODV using the road segment length scenario described in Section G.1. We illustrate in Fig. VI-6(a) how a longer road segment impacts AODV's PDR. As we could expect, RWM is not influenced by longer road seg-

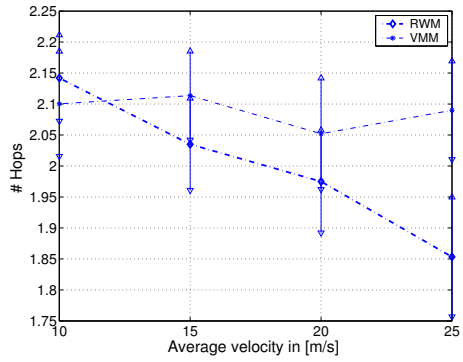
[3]In the remainder of this Chapter, we will refer only to the mobility model for actually mentioning AODV using the mobility model



(a) Packet Delivery Ratio (PDR)



(b) End-to-end delay



(c) Number of hops

Fig. VI-5. Performance evaluation of AODV as a function of the average desired speed

ments. However, AODV's PDR with VMM is significantly improved. Fig. VI-6(b) and Fig. VI-6(c), shows that the length of road segments also influences the delay and the number of hops of AODV. Not only can we see that the average segment length has an effect on the performance of AODV, but also that the difference between VMM and RWM is not negligible. As VMM models more realistic motion patterns than RWM, we expect the performances in Fig. VI-6 for VMM to be closer to reality. Consequently, the length of road segments in urban scenarios should not be neglected. The reason for the bad performance of AODV with RWM comes from the sparseness (average low density) of the network. But as the road segment length increases, the dynamism of the network in urban areas helps AODV to improve its performance. However, as it can be seen in Fig. VI-6(c), this improvement is only local as the number of hops is reduced. Basically, we can deliver more packets, but only to source-destination nodes around a similar intersection.

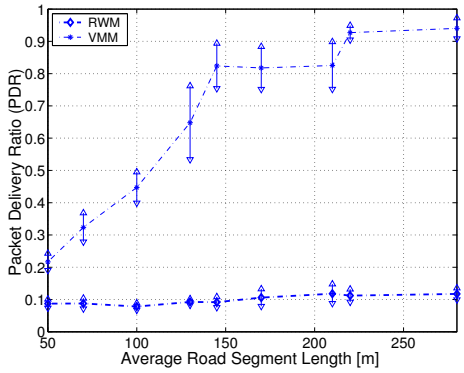
We further carry on the analysis of urban traffic distribution and its effects on AODV using the cluster effect scenario as described in Section G.1. In Fig. VI-7(a), we depict the effect of traffic clusters at intersections, a parameter that does not influence RWM. The PDR is reduced, since it has an impact on the spatial distribution of the vehicles. This observation is corroborated by looking at Fig. VI-7(b), where we see the increasing end-to-end delay, and at Fig. VI-7(c), where the hop count is reduced as the network is only able to deliver data to vehicles in nearby clusters. Again, besides the influence of the parameters on the performances, we see a major performance gap between VMM and RWM. We therefore illustrate how this new parameter is also able to control the performance of AODV for realistic mobility patterns in a way that is not possible by standard parameters. Similarly to the road segment length scenario, the reason for the bad performance of AODV with RWM comes from the sparseness (average low density) of the network. But as cluster effect is increased, the local aggregation around each intersection helps AODV to improve its performance.

G.4 Performances of AODV and OLSR under Vehicular Mobility Patterns

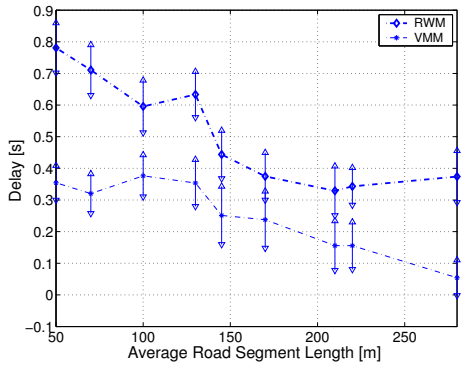
In the previous section, we illustrated how realistic vehicular mobility patterns had a non negligible impact on AODV, as its performance was significantly improved. We can extrapolate that OLSR could also have different performance results under vehicular mobility patterns. We are now therefore interested in conducting a full scale performance evaluation of AODV and OLSR in order to see how they behave in urban environment, and see if conclusions reached in previous studies are still valid.

As we showed in the previous section that the average velocity had no effect on AODV, we only decompose our performance analysis into four different scenarios, where parameters are fixed according to Table VI-4. In the first scenario, we want to see the influence of the average length of road segments. Then, in the second scenario, we analyze the clustering effect at intersections, while in the third scenario, we are interested in the data traffic rate. Finally, in the last scenario, the objective aims at observing the effect of the network density. Each point is the average of 10 samples, while the error bars represent a 95% confidence interval.

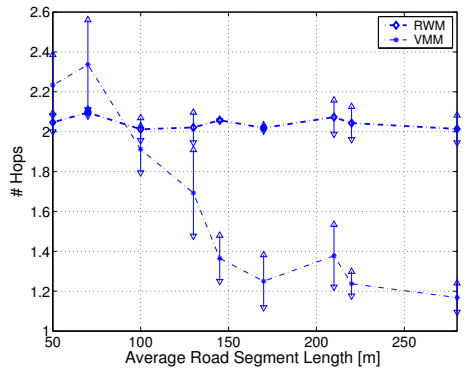
We illustrate, on the first set of simulations, the effect of the average road element



(a) Packet Delivery Ratio (PDR)

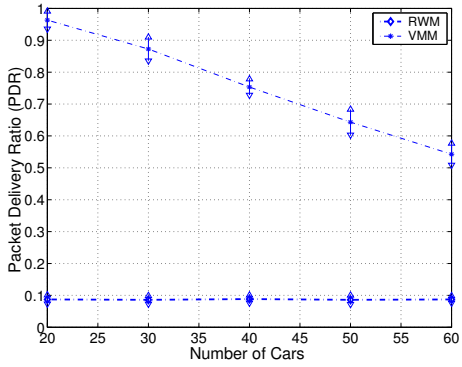


(b) End-to-end delay

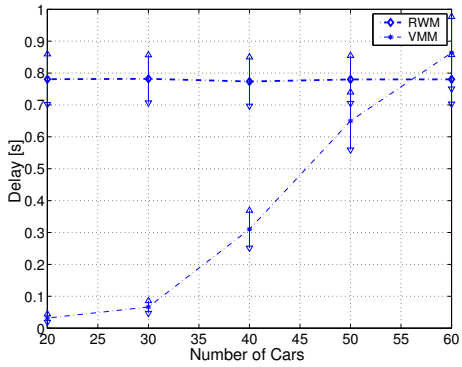


(c) Number of hops

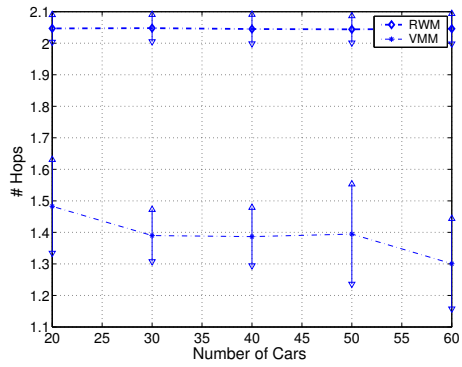
Fig. VI-6. Performance evaluation of AODV as a function of the average length of the roads segments



(a) Packet Delivery Ratio (PDR)



(b) End-to-end delay



(c) Number of hops

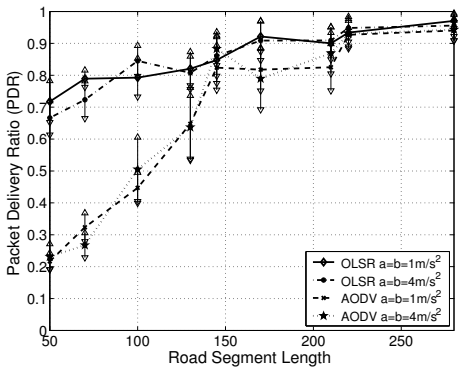
Fig. VI-7. Performance evaluation of AODV as a function of the number of vehicles (cluster effect)

length on the performance of AODV and OLSR using the road segment length scenario as described in Section G.1. On Fig. VI-8(a), we see that OLSR PDR is less sensitive to the road length than AODV's. As we decrease the length of road segments, the distribution of vehicles on the simulation area becomes more and more clustered on intersections, and AODV is more dependent to this effect than OLSR. On Fig. VI-8(b), AODV's control packets drop as the length of road elements increases. AODV RO ends up being 75% lower than OLSR. As we see, OLSR control traffic may be assumed to be independent of the road length, as it is only dependent to network density or velocity. Moreover, the increase in the average speed is too limited to have a direct impact on it. On the other hand, the improved spatial distribution has a major impact on AODV as it improves the dissemination of buffered active routes at intermediate nodes, which in turn reduces the number of control packets required to open a route to a destination vehicle. And as we reduce the amount of control packets to open a route, the delay can also be significantly improved as it can be seen in Fig. VI-8(c), where AODV's end-to-end delay for clustered urban networks is 4 times larger than OLSR's, but ends up being identical for larger road lengths.

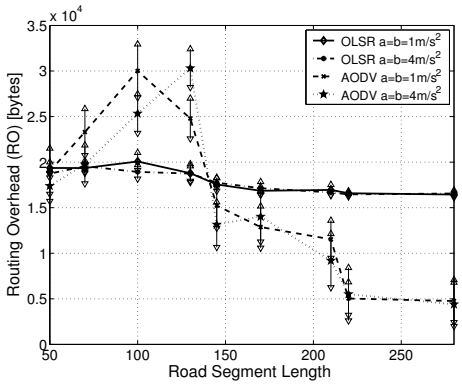
We further carry on the analysis of urban traffic distribution and its effects on AODV and OLSR using the cluster effect scenario as described in Section G.1. On Fig. VI-9(a), we see that neither AODV nor OLSR outperforms the other in term of PDR. Although both protocols are sensitive to urban traffic, OLSR is less dependent to this clustering effect as it accentuates its gap with AODV as the number of vehicles increases. In Fig. VI-9(b), we find a similar results as Fig VI-8(b) where AODV produces less control traffic than OLSR in a non-clustered urban environment, a situation that is reversed for clustered urban environments. Similarly, the AODV's end-to-end delay is significantly increased by an increased clustering effect at intersections.

In both sets of simulations, we however could not see a clear effect of the acceleration, resp. deceleration rate on AODV or OLSR's performance. This comes from the homogeneous distributions of vehicles. Indeed, VMM is not able to model heterogeneous vehicles with different accelerations (a), resp. deceleration (b) rates. And the advantage of an increase (a) or (b) is only beneficial if other vehicles have lower ones. We postpone this analysis to future work.

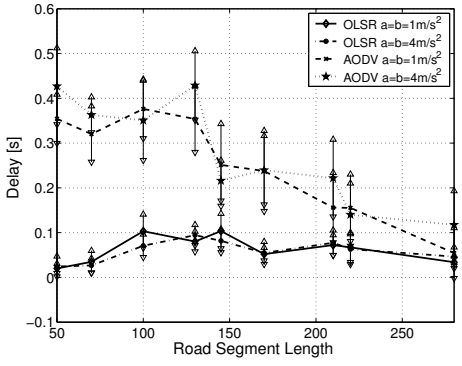
After having analyzed the effect of urban traffic distribution on the performance of routing protocols, we now illustrate the direct influence of data traffic rate and node density (in terms of average number of neighbors per vehicle) on AODV and OLSR performance. As we want to model urban environments, we fix the average road length to 50m and restore the transmission range to 100m. Fig. VI-10(a) shows the average PDR against the CBR throughput. The first observation we can make is that OLSR outperforms AODV on average by 200 %. This is a direct consequence from the previous analysis, which showed that AODV is clearly penalized by the non-uniform distribution of vehicles in the urban environment (see Fig. VI-8(a)). The second observation we can make is that, although both protocols experience a performance decay with the increase of the data traffic rate, the decay is less pronounced for AODV. When the rate of route discoveries is small, so is the probability for intermediate nodes to know an active route to a destination node. Consequently, a large number of AODV route requests (RREQ) must travel up to the destination node. However, as the data rate increases, so does the chance for intermediate nodes to have cached active routes, while OLSR must completely reconfigure its routing tables, a procedure that further restricts the channel



(a) Packet Delivery Ratio (PDR)

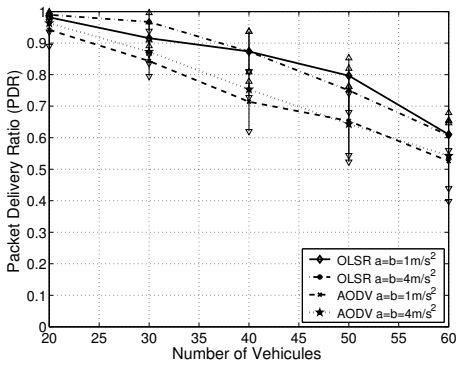


(b) Routing Overhead (RO)

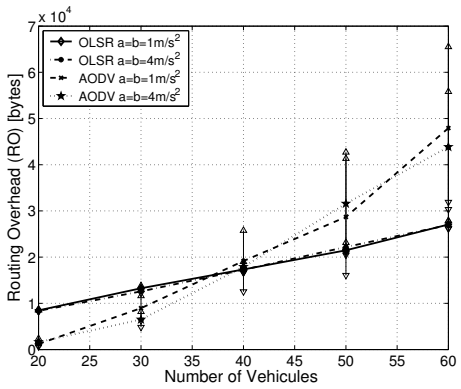


(c) End-to-end delay

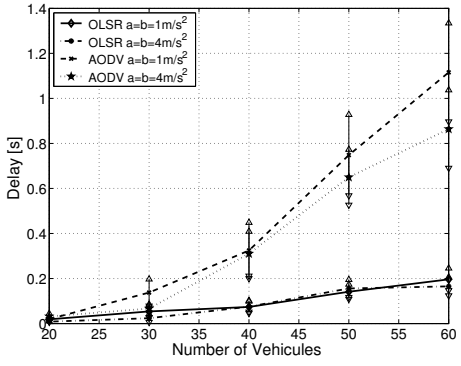
Fig. VI-8. Performance evaluation of AODV and OLSR as a function of the average length of the roads segments



(a) Packet Delivery Ratio (PDR)



(b) Routing Overhead (RO)



(c) End-to-end delay

Fig. VI-9. Performance evaluation of AODV and OLSR as a function of the number of vehicles at a fixed average density (Clustering Effect)

access and reduces active routes for data traffic.

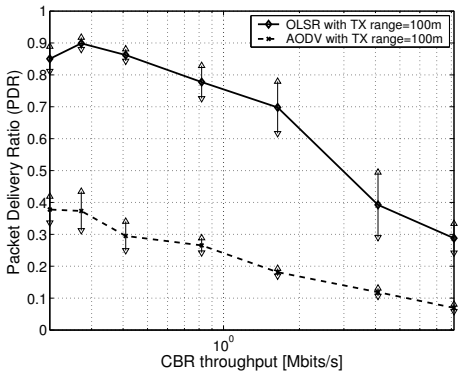
The Routing Overhead (RO) is depicted in Fig. VI-10(b). We actually see that OLSR control traffic is always lower than AODV's, since the cost of repeated route discovery procedures in AODV introduces a large control traffic overhead. Note that this result is consistent with Fig. VI-11(b) as we used a network local density of 12 for this scenario. We also observe that the control traffic of OLSR exhibits the expected characteristics of being independent of the data traffic rate. At very high data rates, the AODV's RO drops significantly, a feature that could be explained by the saturation of the MAC layer.

Finally, we show in Fig. VI-10(c) that OLSR consistently presents the lowest delay, regardless of data traffic. This may be explained by the fact that OLSR, as a proactive protocol, has a faster processing at intermediate nodes. When a packet arrives at a node, it can immediately be forwarded or dropped. In reactive protocols, if there is no route to a destination, packets to that destination will be stored in a buffer while a route discovery is conducted. Accordingly, the performance improvement in terms of delay raises up to 3 times between AODV and OLSR.

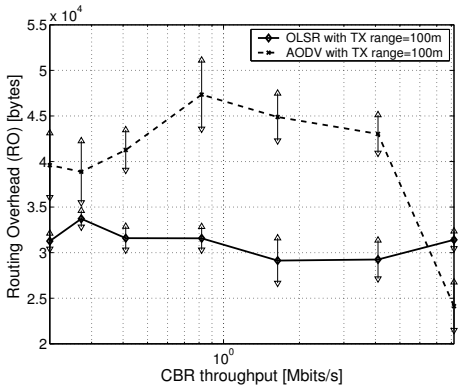
In the next set of figures, we display results obtained for the fourth scenario. Node density is defined as a node's average number of neighbors and is computed as mentioned in Table VI-1. Similarly to Fig. VI-10(a), Fig. VI-11(a) shows that OLSR outperforms AODV by up to almost 300% for highly dense networks. In order to analyze this graph, we divide the graph in three regions: locally *supra-critical, critical, and super-critical* [4] densities. We use the term *locally* because, due to the clustering effect, the network may not be connected even with a high density of nodes. However, within each cluster, *supra-critical, critical, and super-critical* densities appear, which create locally connected components of varying size. In the supra-critical density (8 nbrs/vhcl and below), OLSR is able to benefit from an increasing network density, whereas AODV has a stable PDR. When cars are aggregating in intersections, the MPR nodes become more stable, which increases the stability of OLSR and helps improving OLSR PDR. Then, above a critical density (8 - 10nbrs/vhcl), OLSR's shows initial signs of decrease. Indeed, in the super-critical category, as the density of car locally increases, the periodic maintenance of OLSR reduces its capability of accessing the channel for data traffic, while AODV's RREQ packets have a high chance to find a close intermediate node with an open route. An interesting remark may be made by comparing Fig. VI-9(a) and Fig. VI-11(a). We see on Fig. VI-9(a) that AODV's PDR is penalized by the clustering effect, at a constant network density. Accordingly, AODV is able to improve its PDR as we increase the network density, but the increased cluster effect reduces its performance. As the configurations used to obtain the results displayed in Fig. VI-11(a) include both the influence of the increased number of neighbor and the non-uniform distribution of urban traffic, the effects are mutually exclusive and result to almost stable PDRs.

The next figure depicts the RO of OLSR and AODV as a function of the node density. We can see on Fig. VI-11(b) that, as we would expect, both ROs increase with the density. We clearly see a transition threshold for the control traffic generated by OLSR and AODV. For node densities below 8 nbrs/vhcl, the control traffic overhead of AODV

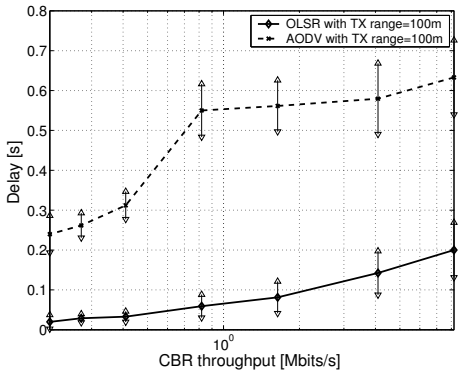
[4] Critical, supra-critical or super-critical are usual terms employed in percolation theory, referring to supra- or super- critical node densities for a network to percolate.



(a) Packet Delivery Ratio (PDR)

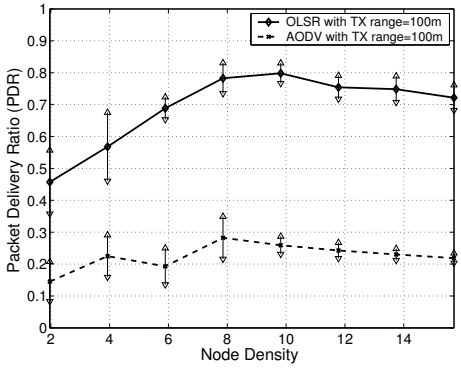


(b) Routing Overhead (RO)

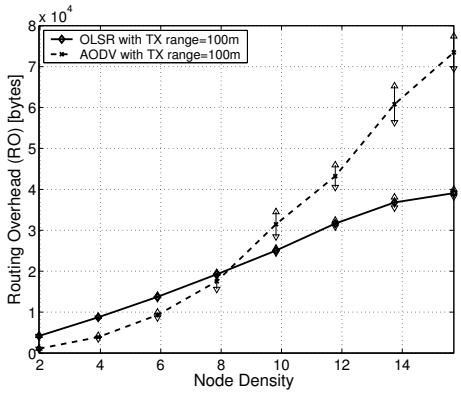


(c) End-to-end delay

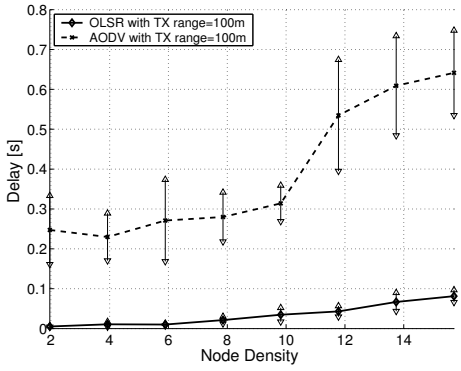
Fig. VI-10. Performance evaluation of AODV and OLSR as a function of Data Traffic Rate



(a) Packet Delivery Ratio (PDR)



(b) Routing Overhead (RO)



(c) End-to-end delay

Fig. VI-11. Performance evaluation of AODV and OLSR as a function of Vehicular Density, where the vehicular density is the average local density measured at intersections

is smaller than OLSR. However, as the density increases, the cost of repeated route discovery procedures in AODV introduces a large control traffic overhead, and OLSR ends up outperforming AODV up to 100%.

Finally, Fig. VI-11(c) depicts the end-to-end packet delay. As the access to the channel becomes harder, the delay can be lowered when a RREQ finds an intermediate node with an active route. However, the penalty for not finding any intermediate node becomes prohibitive as the network becomes locally saturated. On the other hand, routes that OLSR could maintain despite the congested channel are ready to use.

H CONCLUSION

In this Chapter, we first illustrated how vehicular ad hoc networks in urban environment experience particular motion patterns which cannot be properly described by standard parameters. Indeed, the traffic regulations and the vehicles characteristics handled by the *VanetMobiSim Model (VMM)* create a clustering effect at intersection. This effect has remarkable properties on the spatial and temporal distribution of vehicles. The first one is that neither initial nor maximum velocity have a total influence on the real velocity in urban environments. Indeed, due to the interactions with the spatial environment and other neighboring cars, vehicles experience a non negligible speed decay. Then, a second property is the non-uniform distribution of urban traffic which locally increases the density of vehicles, and therefore improve the performance of routing protocol in sparse networks conditions.

As neither the average velocity, nor the average density are able to control the spatial and temporal dependences generated by realistic urban vehicular motion patterns, we defined new meaningful parameters such as the *average length of road segments*, *the acceleration* or *the clustering effect*. By representing the true parameters of the topology or the mobility patterns, we illustrated how they have a significantly larger impact on the performance of AODV.

Another observation is that not only these new parameters are able to remarkably describe urban motions, but also these urban motions actually improve the performances of AODV, as they are significantly increased compared to those with Random Waypoint. These parameters become therefore an important key to more realistic performance evaluations of vehicular ad hoc networks in urban environments. Indeed, the road segment length helps to increase the dynamism of urban networks, while the cluster effect generates "hot spots" at intersections.

We then evaluated OLSR and AODV against urban-specific metrics such as road segment lengths or non-uniform urban traffic distribution, and against regular metrics such as network density and data traffic rate. The obtained results we found showed that the performance of AODV is significantly influenced by the non-uniform distribution of urban traffic that is experienced in urban environments. We showed how OLSR outperforms AODV for almost all performance metrics we used. OLSR may be seen as another possible good candidate for VANETs routing protocols in urban environments. This result is in complete contrast to previous studies, which have either concluded, at best, that reactive protocols were almost identically performing, or even outperforming proactive schemes. The main conclusion from this Chapter is that urban environments

with realistic mobility patterns have a major impact on VANETs routing protocols.

The interest for position-based routing protocols has been growing fast recently. In future work, we plan to investigate the effect of VanetMobiSim on protocols such as the Greedy Perimeter Stateless Routing (GPSR) [11] or the Locally Optimal Routing Algorithm (LORA) [180] in future performance evaluations, as previous works on this field have stated they might be more suited to vehicular networks.

CHAPTER VII

Application of Kinetic Graphs to Broadcasting and Routing

Contents

Abstract—*In this Chapter, we apply Kinetic Graphs to broadcasting and routing in MANETs. We discuss the improvements that multipoint relays may experience by the use of Kinetic Graphs. Multipoint Relaying (MPR) is a technique to reduce the number of redundant retransmissions while diffusing a broadcast message in the network. The algorithm creates a dominating set where only selected nodes are allowed to forward packets. Yet, the election criteria is solely based on instantaneous nodes' degrees. The network global state is then kept coherent through*

periodic exchanges of messages. We propose in this Chapter a novel heuristic to select kinetic multipoint relays based on nodes' overall predicted degree in the absence of trajectory changes. Consequently, these exchanges of messages may be limited to the instant when unpredicted topology changes happen. Significant reduction in the number of messages are then experienced, yet still keeping a coherent and fully connected multipoint relaying network. Finally, we present simulation results illustrating that our approach is significantly better than the MPR algorithm in terms of network coverage, number of multipoint relays, flooding capacity. More interesting, this is even obtained with a drastic reduction of the number of messages exchanged during the process.

Keywords— *Broadcast, Kinetic Multipoint Relay, Mobility Prediction, Mobile Ad Hoc Networks.*

MOBILE Ad Hoc Networks (MANETs) is an emergent concept in view for infrastructureless communication. These networks rely on radio transmissions, but with the lack of infrastructures, flooding (distributing information to each and every node in the network in an uncontrolled way) happens to be a key part of information dissemination. In wireless networks and particularly when the network is dense, the overhead due to this kind of information dissemination may become prohibitive. Despite its simplicity, flooding is very inefficient and can result in high redundancy, contention and collision. This is the main motivation for many research teams that have proposed more efficient flooding techniques whose goal is to minimize the number of retransmissions while attempting to deliver packets to each node in the network. Different approaches of flooding techniques and broadcasting control protocols exist and are listed in [145, 181].

Multipoint relaying (MPR, [182]) provides a localized way of flooding reduction in a mobile ad hoc network. Using 2-hops neighborhood information, each node determines a small set of forward neighbors for message relaying, which avoids multiple retransmissions and blind flooding. MPR has been designed to be part of the Optimized Link State Routing algorithm (OLSR, [4]) to specifically reduce the flooding of TC messages sent by OLSR to create optimal routes. Yet, the election criteria is solely based on instantaneous nodes' degrees. The network global state is then kept coherent through periodic exchanges of messages. Some studies showed the impact of periodic beacons, which could be compared to increasing the probability of transmission, in 802.11 performances [183], or the effects of beaconing on the battery life [184]. This denotes that these approaches have major drawbacks in terms of reliability, scalability and energy consumptions. The next step to their evolution should therefore be designed to improve the channel occupation and the energy consumption.

A SUMMARY OF CONTRIBUTION

In this Chapter, we propose to improve the MPR protocol by using Kinetic Graphs. First, we expose convergence issues for MPR and OLSR which have to be resolved prior to the use of the Kinetic approach. Then, we introduce the *Kinetic Multipoint Relaying (KMPR)* protocol which heuristic selects kinetic relays based on the nodes actual and future predicted nodal degrees. We compare KMPR and MPR with mobility patterns defined by the RWM and VanetMobiSim. We chose to also include the RWM as it will show an upper bound on the performance improvements of KMPR, as

unlike VanetMobiSim mobility patterns, the RWM may be adequately predicted. We show that, in both cases, KMPR is able to improve the flooding reduction at a lower maintenance overhead and more interesting, with a drastically improved delay.

We then propose to study the benefits OLSR may have from the use of KMPR. More precisely, we show that thanks to KMPR's improved topology knowledge, OLSR's Packet Delivery Ratio (PDR) and the Route Error Ratio (RER) are significantly improved. By the low topology maintenance overhead induced by KMPR, OLSR also obtains a better channel access for packet routing.

B Organization of Work

The rest of the Chapter is organized as follows. Section C describes serious convergence issues in the original MPR that need to be solved priorly to adapting Kinetic Graphs. In Section D, we propose to improve the MPR protocol by employing the kinetic graph approach with link weights represented by kinetic nodal degrees. Finally, Section E proposes to study the benefits OLSR may have from the use of KMPR and Section F concludes the Chapter.

C Convergence Issues in MPR

The optimal objective in Kinetic Graphs is to be able to get rid of proactive mobility maintenance beacons. In other words, once the network made the predictions, no further updates are needed. The corollary to this approach is that topology information are not periodically broadcast and that the network must converge in a single logical iteration[1]. The convergence process must therefore follow a predefined sequence of message exchanges and processing where any deviation leads to data inconsistency. In this section, we discuss one particular issue in OLSR that has long been occulted, and which had been a heavy burden to our approach to Kinetic Graph for MPR: *its convergence*.

Authors in [185] claimed that 75% of MPRs were elected on the first MPR step and during the first logical iteration. However, as we will see, this usually requires several physical steps for nodes to obtain correct neighborhood information and accurately elect MPRs. In between, we find suboptimal MPRs, links and neighborhood inconsistency. A more alarming convergence issue, which may occurs even when the election has been correctly performed, is the loss of critical packets. They represent packets containing the logical status of links. This loss, which is due to synchronous transmissions, buffer overflows or other channel considerations, leads to network inconsistency. However, this issue has been occulted in the past by relying on multiple retransmissions and the apparent convergence of OLSR. Yet, we have no guarantee that the routes obtained by OLSR have been build only on correct information. Therefore, OLSR may even not be fully operational for large and dense mobile networks.

We consider convergence as the number of steps needed to make the protocol end.

[1] A logical iteration is defined as an iteration of the targeted protocol (which may include several message exchanges).

Still, we must distinguish logical from physical steps. In order to elect a MPR node, it usually takes 2 logical steps, recursively performed until all two hops neighbors are covered. The physical steps are MPR's ability to notify the elected MPR nodes of their election. Indeed, in a perfect environment, MPR converges after successfully having notified all its MPR nodes of their respective elections. OLSR converges when all TC messages containing the MPR Selectors have been spread on the entire network.

While it is hard to quantify the latter case, simulations we performed showed that the former is very critical. As a matter of fact, in our implementation, we limited the physical steps for MPR election to a single one. Nodes gather information on their neighbors, perform the two steps of the original heuristic of MPR, and notify their respective MPR nodes. In a perfect environment, all this should be far enough to make the protocol converge. Yet, we noticed that packet losses and the order of packet receptions were altering the whole process.

Let us first consider the order of packet decoding. In OLSR RFC3626 [4], upon reception of a packet, a node first considers in that order, Asymmetric links, Symmetric links, MPR links and Lost links, and in the order of the increasing node ID. A typical example of such decoding problem is as follows. Consider the configuration depicted in Fig. VII-1. Depending on the order of the decoding of the message sent by node 2 to node 3, important logical data will be ignored.

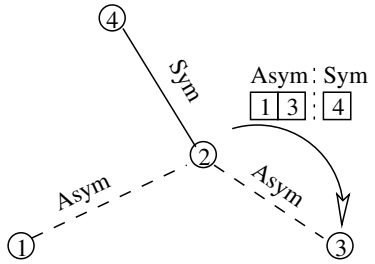


Fig. VII-1. Illustration of OLSR convergence issues

For example,

- Node 3 first decodes the Asymmetric link between node 2 and node 1. Yet, since node 2 is also an Asymmetric neighbor to node 3 and cannot have 2-hop neighbors, this logical status is discarded. The Symmetric logical status of the link between node 2 and node 4 is also ignored if decoded before node 3 decode the Asymmetric link between node 2 and node 3.

We can find several other message discarding problems that are connected to the message decoding order, either within similar or different statuses. Yet, this is more an implementation issue than a MPR misconception. Unfortunately, several implementations of OLSR, including OLSR dæmons, ignore this problem and rely on multiple retransmissions to correct this issue. Consequently, several physical iterations are needed for each node to elect the correct MPRs and reach optimality.

A solution to solve this problem is to first decode the sender status, whether it is Asym-

metrical, Symmetrical, MPR or Lost, then decode the node's own status, and finally decode the rest of the data contained in the message. For example, if node 3 first decodes the status of its link with the sender of the message (node 2), then node 3 is able to extract its 2-hops neighbors logical status and does not discard packets due to inconsistent configurations.

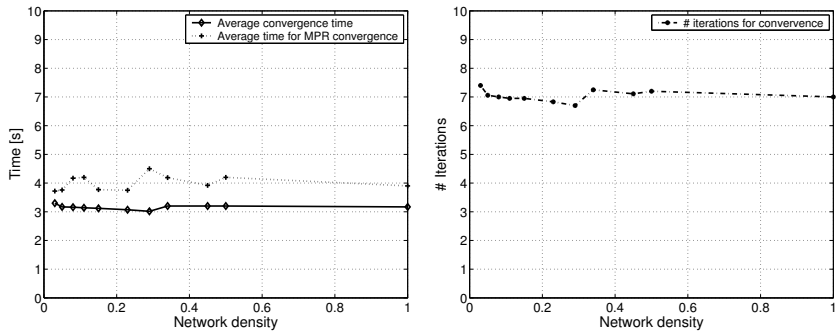
Then, another serious issue that cannot be improved by a particular implementation is network inconsistency due to message losses. We consider here two kinds of message losses for MPR. In order of their importance: *messages containing links physical status*, and *messages containing links logical status*. While the former naturally represents the channel status, the latter is what we call critical packets. Actually, the **weakest link** in OLSR comes from the **strongest link** of MPR. MPR flooding optimality comes from its selective retransmission. However, this is a very critical issue since perfect flooding for MPR and efficient routes for OLSR highly depend on this particular feature. If non-accurate MPR selectors are obtained and propagated due to the losses of those critical packets, MPR nodes will not be coherent with the topology, MPR flooding stops being efficient and OLSR establishes suboptimal routes.

Therefore, incorrect decoding and the losses of critical packets bring serious convergence issues that we depicted in Fig. VII-2. The following results were obtained using the Naval Research Laboratory ns-2 implementation of the OLSR protocol [186]. The following results were obtained by measuring the metrics after the population of 60 nodes were uniformly distributed in a $A \times B$ grid, were A and B depend on the required density of nodes.Each node has a transmission range of $250m$. The density is obtained by the following formula $\#nodes \cdot \frac{\pi \cdot range^2}{A \cdot B}$. We normalized the density with respect to the density of nodes obtained with 60 nodes distributed in a $900m \times 700m$ grid, each one having a $250m$ transmission range. As we want to show convergence issues, we simulated OLSR on a static network without traffic. We are convinced that nodes mobility and traffic will even worsen our results. Finally, the convergence time is defined as the time before all nodes obtain symmetric links to all of their neighbors, while the MPR convergence time is defined as the time before all selected MPR nodes have been correctly notified of their status by all MPR Selector nodes. The number of iterations is similar to the MPR convergence time, but measured in terms of logical iterations.

On Figure VII-2(a), we see that MPR needs on average 3 seconds to converge, before which non-stable MPRs are elected. We also see on the same figure that no stable and optimal MPRs are obtained before 4 seconds on average. Therefore, OLSR cannot expect to create stable routes during this time interval. We also show in Figure VII-2(b) the average number of iterations before MPR converges. We see that MPR needs on average 7 iterations before being able to provide OLSR with accurate topological data. These observations are important since they are obtained based on a static network. If we consider mobility, each time the topology is changed, OLSR looses between 3 to 4 seconds before being able to reorganize its routes[2]. Moreover, these results also show that in highly mobile or heavy loaded networks, OLSR never completely converges as one might expect.

For the analysis performed in this Chapter, we corrected the message decoding issue. However, the loss of critical packets is still an issue to KMPR. Yet, thanks to the Ki-

[2]We do not consider OLSR own link state convergence time which also increases this time



(a) Convergence Time

(b) Convergence Iterations

Fig. VII-2. Illustration of the Convergence of OLSR

netic approach, we reorganize the MPRs less frequently and therefore we lower the dependency to the convergence time.

D KINETIC MULTIPOINT RELAYING

Once the convergence issues described in the previous section had been solved, we were ready to use mobility predictions and Kinetic Graphs on MPR. In this section, we propose to improve the MPR protocol by using Kinetic Graphs. We introduce the *Kinetic Multipoint Relaying (KMPR)* protocol which heuristic selects kinetic relays based on nodes actual and future predicted nodal degrees, called *Kinetic Nodal Degree* introduced in Section E.2. Based on that, periodic topology maintenance may be limited to the instant when a change in the neighborhood actually occurs. Our objective is to show that this approach is able to significantly reduce the number of messages needed to maintain the backbone's consistency, thus saving network resources, yet with similar flooding properties as the regular MPR.

D.1 Short Background on Broadcasting in Mobile Ad Hoc Networks

In [181], the authors propose another designation for broadcasting techniques. They ordered them in four classes *Simple flooding; Probability-based; Area-based and Neighbor knowledge*. While simple flooding, as its name indicates, simply floods packets in the network, the Probability-based approach makes a node forward a packet with a particular probability. The Area-based approach lets a node decide to forward a packet if it brings the best progress in its area, while finally the network-based approach, which MPR belongs to, uses the neighborhood knowledge in order to adequately choose the best relays. This approach has later been adopted by [145] in the author's survey of broadcasting and topology control protocols. This Chapter presents these protocols mainly into two classes: *centralized*, and *localized*; the latter being further differentiated into *Distributed Connected Dominated Set (CDS), Low weighted structures and*

Forwarding approach. Though non scalable, global methods are helpful since localized protocols can be an adaptation from protocols developed for the centralized class. For example, [142] creates a distributed version of the well known minimum spanning tree (MST) obtained in centralized protocols. Finally, a large survey on Dominating sets used to improve broadcasting is provided in [187], and a survey on multipoint relay-based broadcast schemes may be found in [188].

We decided to chose the MPR protocol due to its locality, simplicity and its diffusion, as it is used by the OLSR [4] and is also a candidate for Wireless OSPF [2].

D.2 Basic Idea

Multipoint relaying (MPR, [182]) provides a localized way of reducing the number of redundant retransmissions while diffusing a broadcast message in the network. Using 2-hops neighborhood information, the algorithm creates a dominating set where only selected nodes are allowed to forward packets. MPR has been designed to be part of the Optimized Link State Routing algorithm OLSR [4] to specifically reduce the flooding of TC messages sent by OLSR to create optimal routes. Yet, the election criteria is solely based on instantaneous nodal degrees. The network global state is then kept coherent through periodic exchanges of messages. Some studies showed the impact of periodic beacons, which could be compared to increasing the probability of transmission, in 802.11 performances [183], or the effects of beaconing on the battery life [184]. This denotes that these approaches have major drawbacks in terms of reliability, scalability and energy consumptions. The next step to their evolution should therefore be designed to improve the channel occupation and the energy consumption.

In this section, we propose to improve the MPR protocol 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, periodic topology maintenance may be limited to the instant when a change in the neighborhood actually occurs. Our objective is to show that this approach is able to significantly reduce the number of messages needed to maintain the backbone's consistency, thus saving network resources, and this with a better flooding efficiency than the regular MPR protocol.

We describe next the protocol for the construction of the kinetic backbone. Specifications for the neighborhood discovery and aperiodic maintenance are similar to those described for Kinetic Graphs in Chapter III.

D.3 Kinetic Multipoint Relays

In this section, we describe the Kinetic Multipoint Relaying (KMPR) protocol. It is mainly extracted from the regular MPR protocol. Yet, we adapt it to deal with kinetic nodal degrees.

We recall here the definition of the kinetic nodal degree. Considering $nbrs_i$ as the total number of neighbors detected in node i's neighborhood at time t, we define

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

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. Then, the kinetic degree is obtained by integrating (VII-1)

$$\widehat{Deg}_i(t) = \int_t^{\infty} \left(\sum_{k=0}^{k=nbrs_i} \left(\frac{1}{1+\exp(-a\cdot(t-t_k^{from}))} \cdot \frac{1}{1+\exp(a\cdot(t-t_k^{to}))} \right) \right) \quad \text{(VII-2)}$$

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 2—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 3—Logical Kinetic Degree: The logical kinetic degree is the nodal degree obtained by (VII-2) 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 $N(i)$.

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, 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 $N^2(i)$.

Algorithm 5 Kinetic Multipoint Relaying (KMPR)

Require: Begin with an empty KMPR set.

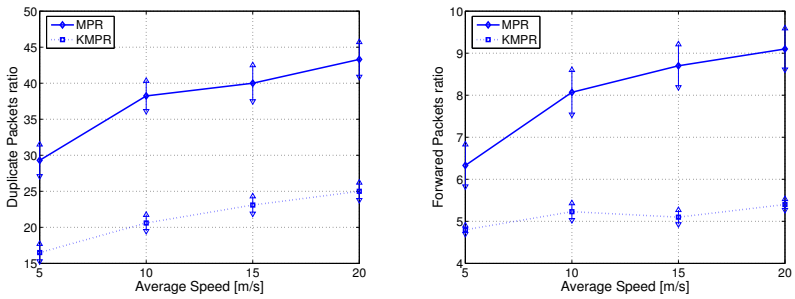
1: Compute the logical kinetic degree of each node in $N(i)$.
2: Add in the KMPR set the node in $N(i)$ that has the maximum logical kinetic degree. Compute the activation of the KMPR node as the maximum covering interval this node can provide. Update all other covering intervals of nodes in $N^2(i)$ 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.

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 4—KMPR flooding: A node retransmits a packet only once after having received the packet the first time from an active KMPR selector.

D.4 Performance Evaluation

We implemented the KMPR protocol under ns-2.29 and used the NRL-OLSR [186] implementation for comparison with KMPR. We measured several significant metrics for MANETs: The effectiveness of flooding reduction, the delay before the network receives a broadcast packet, the number of duplicate packets and finally the routing overhead. Similarly to Chapter VI, we used a square simulation area of 2000×2000 with a node density of $8 nbrs/node$. For the initial results, the mobility model we used is the Steady State Random Mobility Model (RWM) [164], where we made nodes average velocity vary from $5 m/s$ to $20 m/s$. Then, we used VanetMobiSim on the same simulation area and for the same simulation parameters. Please refer to the configuration described in Chapter VI for the configuration of both ns-2 and VanetMobiSim. As we wanted to illustrate the effect of mobility, we did not include pause time between trajectory changes. Finally, we simulated the system for $100 s$ for RWM, and $1000 s$ for VanetMobiSim, where we discarded the first $900 s$ required to reach a pseudo steady state.



(a) Duplicate reception

(b) Forwarding Nodes

Fig. VII-3. Illustration of the flooding reduction of MPR and KMPR for the RWM

Figure VII-3 illustrates the flooding reduction of MPR and KMPR. We can see that even though MPR is already a good algorithm to reduce network flooding, KMPR is able to further improve it. Indeed, the broadcast is reduced by 40% on average compared to MPR. As explained in Section C, we corrected convergence issues of the original MPR algorithm. Therefore, as it converges faster, it is able provide more accurate relays. A second reason comes from the mobility predictions generated by KMPR. It is indeed able to proactively maintain the MPR nodes with respect to mobility.

On Figure VII-4, we depicted the broadcast efficiency of MPR and KMPR. In the simulations we performed, we measured the broadcast efficiency as the time a packet takes before being correctly delivered to the entire network. As we can see, KMPR

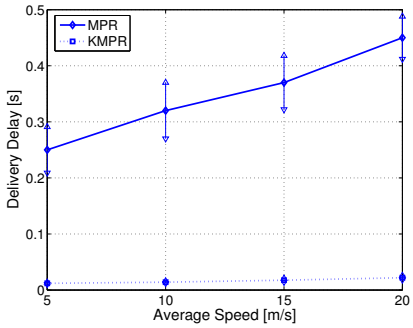
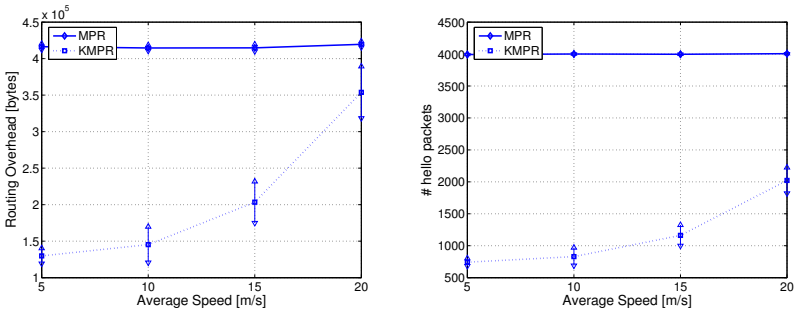


Fig. VII-4. Illustration of the broadcast efficiency of MPR and KMPR for the RWM

has a delivery time 10 times faster than MPR. Similarly to the flooding improvement, as KMPR converges faster and needs less maintenance duty cycles, optimal MPRs are always available. Morevoer, as we will see in the next figure, KMPR's backbone maintenance is significantly less than MPR. Therefore, the channel access is faster and the probability of collisions is decreased.This is a significant illustration of the benefit of Kinetic Graphs to the MPR protocol.

In the two previous Figures, we have shown that KMPR could significantly improve the flooding reduction as well as the broadcast delay. Now, in Fig. VII-5, we illustrate another benefit of KMPR: its *low maintenance overhead*. Indeed, since KMPR uses mobility predictions and does not rely on periodic maintenance, the routing overhead may be reduced by 70% as it may be seen on Fig. VII-5(a). We also show on Fig. VII-5(b) the number of hello messages which drops dramatically with KMPR, yet still preserving the network's consistency. We can moreover see that, as mobility increases, the predictability is reduced and reduces the performance of the predictions. Accordingly, KMPR maintenance overhead tends to converge to MPR's, as Kinetic Graphs degenerate to static graphs when predictions are either wrong or impossible to obtain.



(a) Maintenance overhead

(b) Number of Hello packets

Fig. VII-5. Illustration of the network load for MPR and KMPR for the RWM

Figure VII-6 illustrates the *Unconnected Dominating Set Ratio*. All results obtained so far has 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). Yet, as one can imagine, mobility and maintenance traffic either spatially or temporally disconnect the Connected Dominating Set created by MPR. This figure therefore illustrates the ratio of runs which showed a disconnected graph with respect of the total number of runs. It is also straightforward from this figure to see that, as KMPR is less sensitive to mobility and has a reduced channel occupation, these temporal or spatial disconnections are significantly reduced, further improving the reliability of KMPR.

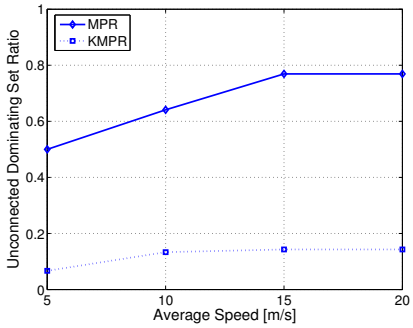


Fig. VII-6. Illustration of Unconnected Dominating Set Ratio with respect to Mobility for the RWM

All Figures we have presented yet have been obtained using the regular Random Waypoint Mobility model. We illustrated in Chapter V and Chapter VI its lack of realism on Mobility Patterns and also on Routing Protocols. In order to further analyze KMPR using more realistic mobility scenarios, we also simulated KMPR and MPR using vehicular patterns. As the RWM may be adequately predicted, results for RWM may be seen as upper bounds (resp. lower bounds depending on the criterion), whereas predicting VanetMobiSim motion patterns is much harder, thus leading to reduced performance.

In Fig. VII-8, we depict KMPR and MPR's behaviors under a vehicular mobility model. Similar to the previous Figures, Fig VII-8(a) shows the drastic reduction of KMPR's backbone maintenance, as well as KMPR's improved broadcast delay. KMPR reduces the maintenance overhead on average by 40%, while the broadcast delay is reduced by a factor of 10 times. The flooding reduction illustrated in Fig VII-8(c) and Fig. VII-8(d) is also improved, even though the confidence intervals are large. Indeed, the broadcast efficiency is subject to a non negligeable variance mostly due to the particular urban topology. Indeed, as we can expect from the spatial distribution illustrated in Chapter V, the broadcast efficiency will drastically change if the broadcasting vehicle is in a clustered intersection or on a road segment, as the density and the velocity impact it.

Finally, we again depict in Fig. VII-7 the unconnected dominating set ratio. As mentioned before, due to the particular spatial distribution of cars creating clusters at intersections and a sparse connectivity in between, frequent disconnections occur for MPR. KMPR is also subject to this disconnection but is able to adapt to dynamic configurations much faster and therefore improves the stability of its connected dominating set.

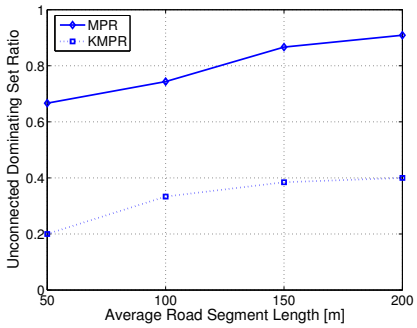


Fig. VII-7. Illustration of Unconnected Dominating Set Ratio with respect to Mobility for the VanetMobiSim

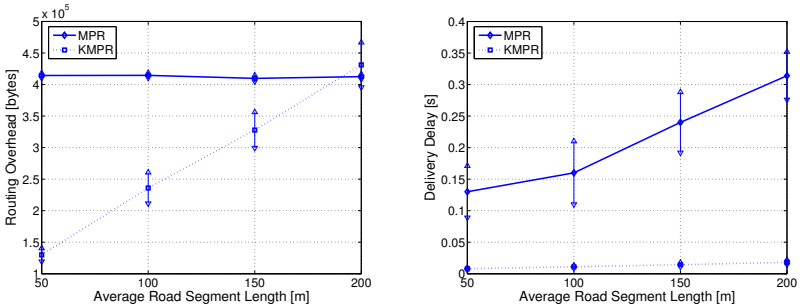
In this section, we have illustrated how MPR could be successfully improved by the use of Kinetic Graphs. We also showed that even though the prediction schema is not in adequacy with the vehicular motion patterns, the use of the Kinetic Degree instead of a Kinetic Distance is able to limit the scope of the prediction error. KMPR is therefore also efficient when used on vehicular motion patterns.

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.

In the next section, we will use the OLSR protocol on top of KMPR, and will see how OLSR is able to benefit from KMPR's improved backbone maintenance for the maintenance and efficiency of its routing tables.

E Application of KMPR to OLSR

In this section, we propose to study the benefits OLSR may have from the use of KMPR. More precisely, we show that thanks to KMPR's improved topology knowledge, OLSR's Packet Delivery Ratio (PDR) and the Route Error Ratio (RER) are significantly improved. By the low topology maintenance overhead induced by KMPR, OLSR also obtains a better channel access for packet routing. However, as results will show, the collaboration of OLSR and KMPR is also victim of its own success. Indeed, since the packet dropped rate is reduced, a significantly larger number of packets are buffered for transmission after channel access. And it is of public notoriety that the 802.11b MAC protocol does not scale either with the number of nodes or the size of uncoordinated traffic. However, the channel access and contention is not in the scope of this work and we will only consider the routing features in this Chapter. Our objective is to show that after being under study for MPR, mobility prediction is also able to improve OLSR.



(a) Maintenance Overhead

(b) Delay

(c) Forwarding Nodes

(d) Duplicate reception

Fig. VII-8. Illustration of the broadcast efficiency of MPR and KMPR under vehicular mobility

E.1 Basic Idea

Although deeply interleaved together, OLSR and MPR have two independent tasks. The role of MPR is to provide a optimized flooding control mechanism, while OLSR's task is to create routing tables. Both protocols have their own comparison criteria and can be independently compared and improved. We can imagine to use OLSR with a different flooding reduction algorithms, or make AODV benefit from MPR to reduce the diffusion of *RREQ* messages.

As a matter of fact, the research community interested in improving OLSR already successfully tested it with different flooding reduction algorithms, such as NS-MPR, S-MPR, MPR-CDS, and E-CDS [189, 190, 191]. They all reached the same conclusion that although creating a larger set of relays, the original MPR protocol reaches a higher broadcast throughput than other tested flooding control algorithm and is better suited for OLSR.

This was our motivation for using MPR in order to adapt the mobility prediction technique and also a justification for testing KMPR with OLSR as we do in this Chapter.

E.2 KMPR applied to OLSR

In order to construct and maintain its routing tables, OLSR periodically sends link state information in the network. The interaction between OLSR and MPR is therefore that OLSR benefits from *MPR flooding* to reduce the redundant transmission of identical TC packets (also known as the Broadcast Storm Problem).

KMPR creates a set of KMPR selectors and their respective activations. Compared to MPR, the difference is that KMPR has computed actual and future KMPR selectors. Each KMPR selector and its relaying capability will be activated when its activation becomes valid.

Accordingly, we can see that OLSR can be easily adapted to use KMPR instead of MPR. It will still periodically send topology messages and the forwarding decision is simply kept transparent to it. Indeed, each OLSR TC message is forwarded by KMPR according to Definition 4. Although KMPR uses activations in order to maintain its set of KMPR selectors, each forwarding decision will be taken by each node based on Fig VII-9.

Upon convergence, KMPR provides OLSR with a table of ready-to-use KMPR selectors. These are similar to MPR selectors with the exception that they are not periodically elected, but are valid for a time interval. Although KMPR is a per-event protocol and no periodic maintenance is performed, OLSR still keeps its periodical feature. Accordingly, OLSR will simply consider KMPR Selectors as simple MPR Selectors and will propagate them in the network using TC messages. No particular modifications are needed for OLSR to run on top of KMPR.

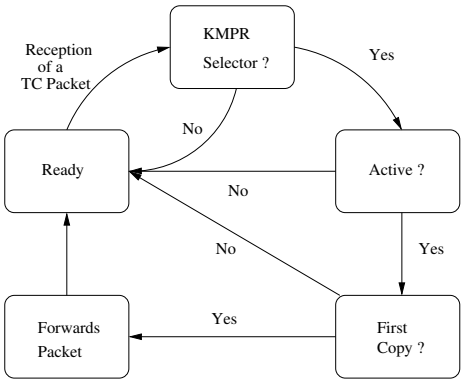


Fig. VII-9. Illustration of the fowarding decision of KMPR

E.3 Performance Evaluation

We implemented the OLSR-KMPR protocol under ns-2 and compared it with OLSR-MPR. The global parameters we used for the simulations are given in Table VII-1. We measured several significant metrics for MANETs routing:

- *Packet Delivery Ratio (PDR)*– It is the ratio between the number of packets delivered to the receiver and the expected number of packet sent.
- *Route Error Ratio (RER)*– It represents the ratio between the number of packets dropped due to the lack of valid routes, and the total number of packet sent.
- *Routing Overhead Ratio (ROR)*– It represents the ratio between the number of routing bytes and total number of bytes correctly received.
- *Delay*- It measures the average end-to-end transmission delay.

Finally, we decomposed our performance analysis in three different scenarios, were we fixed the parameters according to Table VII-2. In the first scenario, we want to see the influence of an increased data rate, whereas in the second scenario, the objective is to test the influence of network mobility.

Figures VII-10 and VII-11 illustrate the Route Error Ratio (RER) of OLSR. The route error ratio represents the ratio between the number of packets which could not find a correct route and the total number of packet sent. We can see that OLSR-KMPR managed to only have 6% of route errors, while OLSR cannot go below 14%. As expected, due to the increased channel access, the route errors are bigger when more sources are sending CBR traffic in the case of OLSR. Since KMPR requires less channel access to maintain its backbone, it is then less penalized when the channel is saturated. Averaged on the CBR rates the sources, the route error ratio is 14% for OLSR and 7% for OLSR-KMPR, which is 2 times less than the regular OLSR. This feature is due to the improved topology knowledge KMPR is able to maintain. Thanks to mobility prediction, KMPR always knows where its neighbors are, thus is able to keep accurate neighborhood information. Consequently, KMPR is able to provide OLSR with more stable and reliable routes.

Network Simulator	ns-2.29
OLSR Implementation	NRLOLSR [186]
Simulation time	100s
Simulation Area	2000m x 2000m grid
Tx Range	250m
Mobility Model	Steady State RWM
Node Speed	Uniform
Network Density	$\#nodes \cdot \frac{\pi \cdot range^2}{X_{dim} \cdot Y_{dim}}$
Data Type	CBR
Data Packet Size	512 bytes
MAC Protocol	IEEE 802.11 DCF
MAC Rate	2 Mbits/s
Confidence Interval	95%

TABLE VII-1. SIMULATION PARAMETERS

Scenarios	Data Rate	Network Mobility	Nodes Density
Data Rate	0.08 Mbits/s to 2 Mbits/s	10 m/s	8.7
Network Mobility	0.8 Mbits/s	5m/s to 15m/s	8.7

TABLE VII-2. SIMULATION SCENARIOS

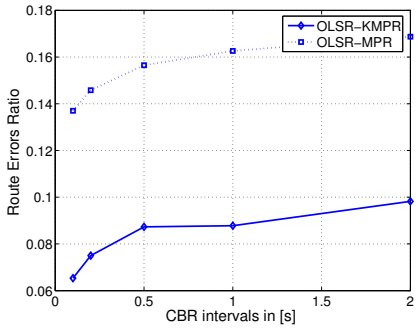


Fig. VII-10. Illustration of the Route Error Ratio given the CBR rate with 10 CBR sources

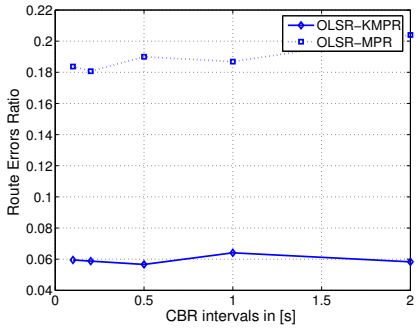


Fig. VII-11. Illustration of the Route Error Ratio given the CBR rate with 20 CBR sources

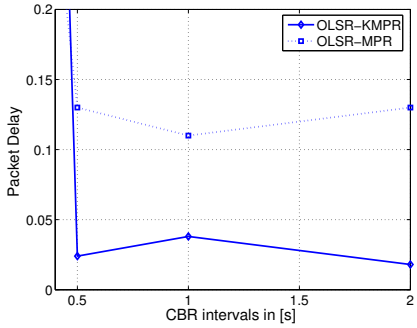


Fig. VII-12. Illustration of the Routing delay given the CBR rate for 10 CBR sources

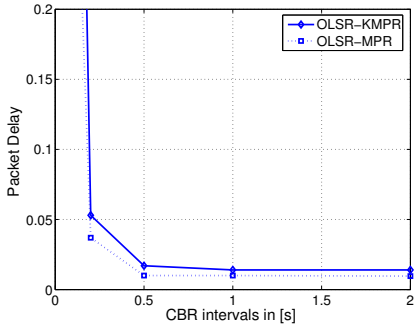
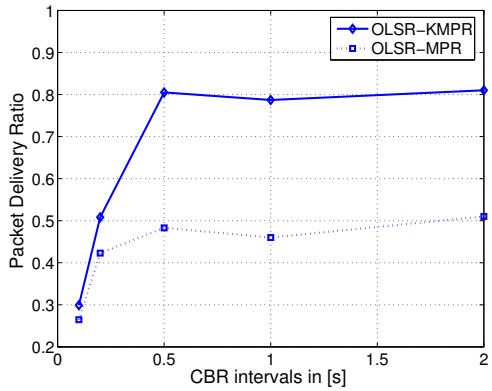
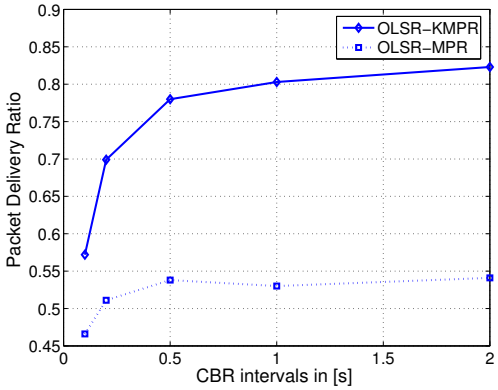


Fig. VII-13. Illustration of the Routing delay given the CBR rate for 20 CBR sources

As we could expect from results in Section D.4, the low broadcast delay of KMPR should impact on OLSR routing. We can clearly see this effect on Fig VII-12 when only 10 CBR sources are used. OLSR is able to obtain an average 80% reduction of the Packet Delivery Delay for low throughput. Yet, either when the CBR throughput or the number of CBR sources saturates the network as in Fig VII-13, the routing delay is slightly bigger for OLSR-KMPR than for regular OLSR. However, unlike OLSR, this delay is mostly generated by saturated relaying queues. One reason for this effect may be that, as the broadcast is reduced, coordinated unicast traffic like CBR is able to be transmitted more efficiently on the OLSR or OLSR-KMPR routes and saturating the transmitting queues.



(a) 10 CBR sources

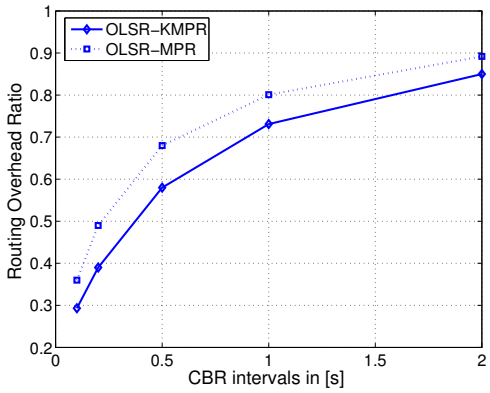


(b) 20 CBR sources

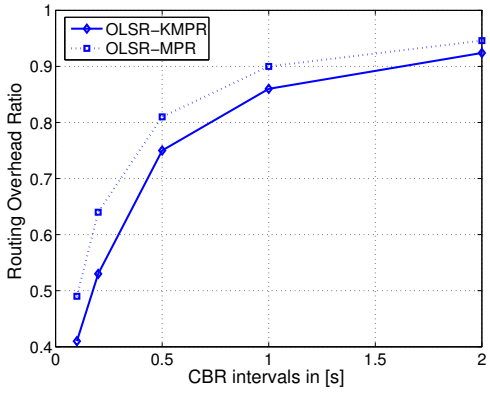
Fig. VII-14. Illustration of the Packet Delivery Ratio given CBR Traffic

In Fig. VII-14, we depicted the main results of this work, that is the improved Packet Delivery Ratio (PDR) of OLSR-KMPR compared to regular OLSR. The packet delivery ratio (PDR) is the ratio between the number of packets delivered to the receiver with

the expected number of packet it should have received, which is a fair measurement of a protocol efficiency. We can see on Fig. VII-14 that the PDR of OLSR-KMPR is improved compared with OLSR. This figure shows that by using mobility predictions, OLSR-KMPR manages to have an average packet delivery ratio increased by 50%. The packet delivery ratio is also not influenced by increased CBR sources or rates. However, as we can see for high CBR throughput, both OLSR and OLSR-KMPR suffer from a dramatic drop of PDR when the CBR rate increases above a certain threshold. However, this particularity is not linked to the routing capabilities of those protocols, but to the wireless channel access limitations.



(a) 10 CBR sources

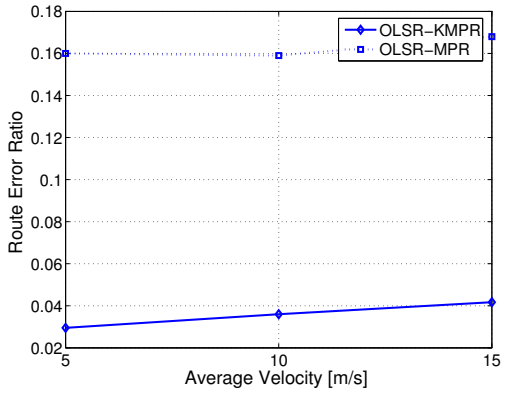


(b) 20 CBR sources

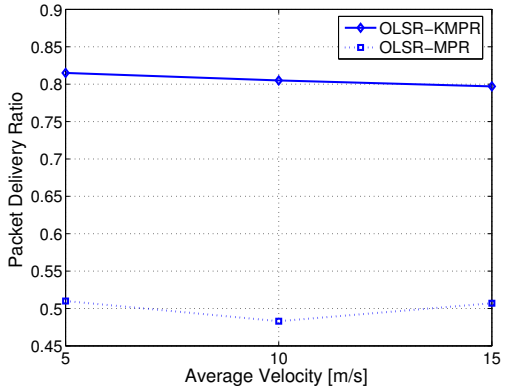
Fig. VII-15. Illustration of the Routing Overhead Ratio given CBR Traffic

Figure VII-15 shows the Routing Overhead Ratio (ROR) induced by OLSR-KMPR and the regular OLSR. The routing overhead ratio is the ratio between the routing packets and total number of packet sent on the network. It represents the cost of using a particular protocol for routing in ad hoc networks. As we mentioned in the Introducing

section, OLSR-KMPR is able to improve OLSR properties at virtually no extra cost. Fig. VII-15(a) and Fig. VII-15(b) are the illustration of this argument. Indeed, we can see that the routing overhead of OLSR-KMPR is less than that of the regular OLSR, yet maybe not as high as expected. The reason is that even though OLSR-KMPR has a lower maintenance overhead, it also transmits more traffic as we illustrated in the previous paragraphs. Therefore, the routing overhead ratio is reduced. However, we must put this effect in perspective to the improved Packet Delivery Ratio.



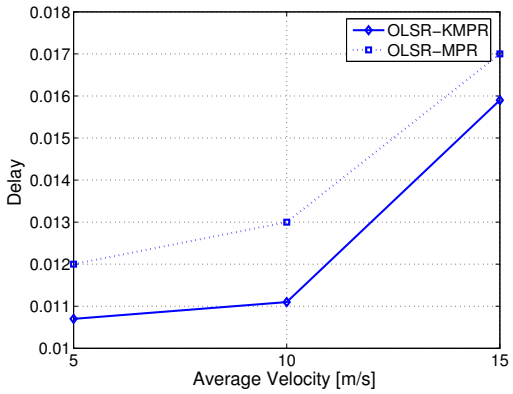
(a) Route Error Ratio



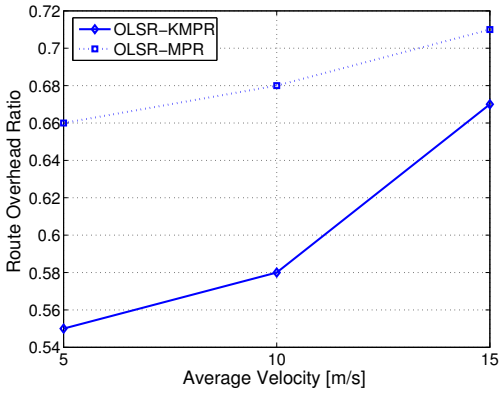
(b) Packet Delivery Ratio

Fig. VII-16. Illustration of the performance of OLSR-KMPR and regular OLSR given the velocity for the RWM

Finally, as we are testing the performance of mobility prediction for OLSR, we finally test OLSR-KMPR for different mobility values. In Fig. VII-16 and Fig. VII-17, we illustrate the effect of mobility on the previous performance criteria. Although the RER and the PDR seems not to be significantly influenced by nodes mobility, the routing delay and the routing overhead ratio are. However, this behavior is not particular to the



(a) Route Delay



(b) Routing Overhead

Fig. VII-17. Illustration of the performance of OLSR-KMPR and regular OLSR given the velocity for the RWM

use of mobility predictions. However, what is particular to mobility prediction is that OLSR-KMPR performs always better than the regular MPR under various mobility scenarios. And it is particularly true for route errors and packet deliveries. The RER is 5 times smaller than regular OLSR, while the PDR is increase by 60%.

F CONCLUSION

In this chapter, we applied an original approach for applying mobility predictions to Mobile Ad Hoc Networks (MANET) called the Kinetic Graphs. The objective was to construct and maintain a topology or routing structures without relying on periodic maintenance. The approach is independent of the criteria used in order to build the backbone, and various approaches may be tested. Accordingly, we developed a criterion less sensitive to inaccurate prediction models called the *Kinetic Degree* and adapted the Kinetic Graph approach to the Multipoint Relaying protocol (MPR).

A major issue in Kinetic Graph which is independent of the metric or the prediction method is the need for protocols to converge in a single logical iteration as no extra beacons will be sent in case information is lost due to improper handling of messages at the protocol side. This was a major setback when we designed KMPR, as MPR showed serious convergence issues which we had to correct before being able to obtain a stable backbone using a minimum number of messages.

We first illustrated the convergence issues of OLSR by observing that MPR needs on average 3 seconds to obtain symmetric links to all its neighbors, and cannot compute stable MPRs before 4 seconds on average. In number of iterations, this mean that MPR needs at least 7 iterations before being able to provide OLSR with MPR selectors. We later proposed solutions to correct this issue, including selective decoding and adaptive retransmission techniques.

Then, we presented an promising approach for improving the well-known MPR protocol by using mobility predictions. We showed that the Kinetic Multipoint Relaying (KMPR) protocol was able to meet the flooding properties of MPR, and this by reducing the MPR maintenance overhead by 70% and MPR broadcast delay 10 times faster under random motions.

We furthermore tested the KMPR protocol under realistic vehicular motions, and could also show that KMPR managed to reduce the channel access by 60% for vehicular motion. At the same time, the broadcast delay could be reduced by 25%, and this with an improved flooding reduction with respect to MPR.

Then, we presented a study of the application of Mobility Predictions to the OLSR protocol. We showed that OLSR packet delivery ratio may be improved by a factor of 50% to 60%, while the route error can be between 2 and 6 times smaller. More interesting, these improvements are obtained at virtually no extra cost since OLSR-KMPR routing overhead ratio is smaller that OLSR. We consequently illustrated that, after having been studied in other fields of mobile ad hoc networking, mobility predictions are also an interesting technique to improve proactive routing protocols, and that more specifically, OLSR performances may be significantly improved by the use of KMPR and Kinetic Graphs.

There are other advanced MPR algorithms, such as NS-MPR, S-MPR, MPR-CDS, or E-CDS (see [188]). It could be interesting as future work to see the comparisons between KMPR-OLSR and these advanced MPR-OLSR solutions. Another future promising work is to further reduce any periodic maintenance and try to use the Kinetic Multipoint Relays and their activations in order to suppress periodic TC messages. In such approach, that could be named *Kinetic Link State Routing (KLSR)*, Kinetic Graphs will have been totally adapted, as no periodic maintenance, either from MPR or OLSR, will be required for efficient routing in MANET.

CHAPTER VIII

Conclusion

Wireless Ad Hoc Networks are an extreme configuration of wireless networks, which do not relay on any fixed or wired infrastructure, and where terminals are self-configuring in order to provide distributed multi-hop wireless communications. The lack of infrastructure or coordinator favors chaotic situations generating a large waste of critical resources.

Creating and maintaining a structure in an ad hoc network requires localized and distributed approaches. Many different propositions have been done in order to build an efficient backbone only based on local information. However, almost none of them explicitly considered the mobility of the nodes part of the backbone.

Yet, mobility is a major source of burden for the maintenance of the structure. Indeed, localized protocols are not sufficient in order to efficiently maintain structures in mobile wireless ad hoc networks, as mobility makes this structure adapted to past configurations only. Therefore, a periodic maintenance has to be created in order to adapt the backbone to the mobile topology. Moreover, depending on the dynamic of the network, the local maintenance becomes resource demanding.

This Doctoral Thesis focused on studying and improving mobility management in ad hoc networks using *Kinetic Graphs*. This approach employs efficient localized methods as well as kinetic structures in order to efficiently construct and maintain a communication backbone in mobile wireless ad hoc networks. We studied the strong required interaction between modeling and predicting mobility in order to efficiently adapt Kinetic Graphs to wireless ad hoc broadcasting and routing protocols, even on the very challenging vehicular networks.

A OUTLOOK OF THE WORK

We presented an original approach for applying mobility predictions to Mobile Ad Hoc Networks (MANET) called the *Kinetic Graphs*. The objective was to construct and maintain topologies or routing structures without relying on a periodic maintenance. The approach is independent of the criteria used to build the backbone, thus various approaches could be tested.

The Kinetic Graph approach is a framework constituted of a neighborhood discovery phase, of a trajectory estimation phase, of a graph construction based on a time varying link weight, and finally of an aperiodic neighborhood maintenance. During the neighborhood discovery phase, we chose to transmit geo-localization information as a mean to generate the mobility predictions. However, the periodic transmission of those information generates a non negligeable overhead, which we reduced by the introduction of a geo-localization compression method.

We then described two possible link weights as criteria for the graph constructions: *Kinetic Distance* and *Kinetic Nodal Degree*. While the former is able to generates energy efficient topology control graphs, the latter is more resilient to prediction errors.

Mobility prediction errors actually depends on four criteria: *realism*, *adequacy*, *predictability*, and *similarity*. We showed how Kinetic Graphs were able to reduce the prediction errors simply to *realism* and *adequacy*, at the cost of a performance related to the *predictability*. Therefore, in order to reduce these errors, we first neglected realism and generated a prediction model in total adequacy with the Random Waypoint Model (RWM). Yet, the efficiency of Kinetic Graphs still depends on the predictability of the motion patterns, in our case those obtained from the RWM.

We therefore performed a study of this predictability and illustrated how the RWM, paradoxically, had a predictability of $\approx 30s$ for reasonable configuration motion parameters. This outcome justified to further study the benefit of Kinetic Graphs in MANETs.

Accordingly, we developed the *Kinetic Degree* and adapted it to the Multi-point relaying protocol. The *Kinetic Multipoint Relaying (KMPR)* protocol uses an heuristic which selects kinetic relays based on the nodes actual and future predicted nodal degrees. Based on that, periodic topology maintenance may be limited to the instant when a change in the neighborhood actually occurs. We showed how this approach was able to significantly reduce the number of messages needed to maintain the backbone's consistency, thus saving network resources, with an improved flooding efficiency compared to the regular the MPR protocol. We also showed how the OLSR could positively benefit from KMPR in order to improve routing efficiency.

Although being motivating, it was also clear that the prediction model was able to bring significant improvements, not only due to the prediction algorithm itself, but also due to the lack of realism of random models. Indeed, vehicles for instance, have specific movements not well described by actually mobility models. Another important aspect we illustrated between modeling and predicting mobility was the relationship between prediction schema and realistic motion patterns. If a prediction model fits to a mobility model which is not realistic, it will not be able to accurately predicts realistic motion patterns. Yet, realism depends on the application.

We therefore chose to adapt our prediction schema to vehicular mobility for various reasons, among them was the improved predictability of vehicles, the availability of GPS coordinates and synchronization, or simply to the wide range of applications based on vehicular motions. As available mobility models were not able to generate realistic vehicular traffic, we created *VanetMobiSim*, a freely available generator of realistic vehicular movement traces for telecommunication networks simulators. VanetMobiSim has been validated first by illustrating how the interaction between featured macro- and micro-mobility was able to reproduce typical phenomena of vehicular traffic. Then, the traces generated by VanetMobiSim were formally validated against those obtained from CORSIM, a benchmark traffic generator in transportation research.

We moreover depicted how the realistic motion patterns created by VanetMobiSim affected the node's velocity, and how new parameters became necessary to evaluate the performance of routing protocols in Vehicular Ad Hoc Networks (VANETs), and evaluated the performance of two ad hoc routing protocols, AODV and OLSR, in realistic urban scenarios.

Finally, we used the realistic vehicular motion patterns on KMPR. Even though we expected the first order prediction model used by KMPR not to be sufficient to accurately predict complex vehicular mobility, we proved that it was not the case and Kinetic Graphs were able to significantly improve the MPR and OLSR protocols even with vehicular motion patterns.

This last outcome concluded our research on modeling and predicting mobility in wireless ad hoc networks, as we showed how by realistically modeling, then devising an efficient mobility prediction schema, we could successfully adapt the Kinetic Graph approach to ad hoc broadcasting and routing protocols, and reduces the important effect of mobility on their performance.

B FUTURE WORK

Based on the results we obtained in this Doctoral Thesis, significant new research orientations could be envisioned. First, as described in this work, the efficiency of prediction schemes for Kinetic Graphs depends on the adequacy between the kinematic model and the motion patterns. In this thesis, we only used a first order kinematic model, yet we employed Kinetic Degrees in order to limit the prediction errors. However, as illustrated in Chapter II, powerful prediction schemes have been devised in the past, and which could be successfully applied to ad hoc networks. Based on a particular efficient prediction schema, it would be interesting to analyze the predictability of vehicular motion patterns. We could therefore reach the optimal Kinetic Graph behavior with a perfect adequacy, a perfect realism, and a large predictability.

In this Doctoral work, we provided two time varying link weights which could be used as criteria for building kinetic graphs. We developed KADER based on the kinetic distance, and KMPR based on the kinetic degree. Yet, other solutions may be foreseen. For example, the Localized Minimum Spanning tree creates a structure based on local and distributed spanning trees. As each node builds a Euclidian Spanning Tree, we could use the Kinetic Distance in order to create a Kinetic Localized Spanning Tree (KLMST). Or, more generally, it would be interesting to adapt the Kinetic Degree to

any ad hoc protocol using nodal degrees as a decision criterion. We illustrated how both criteria could be successfully adapted to wireless ad hoc networks, but we also think that the Kinetic Graph approach also has promising potentials in other fields.

During our research work, we also developed a realistic vehicular mobility model and validated it against a benchmark traffic generator.A very important factor when simulating highly mobile networks is the radio propagation model. Results obtained without accounting for the impact of large obstacles, such as buildings, on the radio signal propagation can hardly be realistic. Studying this aspect would also be challenging, as it would be possible to benefit from the availability of a detailed topology description to introduce a new component in VanetMobiSim, for instance, capable of generating radio propagation information for network simulators.

Moreover, VanetMobiSim is at this time not capable of interacting with a network simulator. As vehicular applications are purposely targeted at altering trajectories based on unexpected events, such as accidents, traffic jams, or simply advertisement, the network simulator and the mobility simulator should be closely interlinked.

Finally, a more general consideration is that the study of mobility is a challenging task that is impossible to completely fulfill. Engineers and researchers always inspired themselves from Mother Nature for solutions to human problems. For example, the Neural Network research field aims at trying to synthetically recreate the behavior of neuronal cells to improve artificial intelligence. Human beings are evolving in a constantly moving environment, and even without knowing it, they perform basic mobility predictions on a daily basis. For example, they predict their mobility and that of others when they try to find a short cut on their daily compute, or more dangerously as a way to drive closer to neighboring cars. Mobility prediction is a natural reflex and studying it could provide solutions to otherwise unsolvable problems.

Bibliography

[1] C. Gentile, J. Haerri, and R. E. Van 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. [11, 43, 68, 69]

[2] "Wireless open shortest path first," `http://hipserver.mct.phantomworks.org/ietf/ospf`. [23, 175]

[3] E. Baccelli *et al.*, "Ospf mpr extension for ad hoc networks," February 2007, Internet Draft, `http://www.ietf.org/internet-drafts/draft-baccelli-ospf-mpr-ext-03.txt` (work in progress). [23]

[4] P. Jacquet *et al.*, "Optimized link state routing protocol (olsr)," Request for Comments, `http://www.ietf.org/rfc/rfc3626.txt`. [23, 74, 78, 99, 112, 170, 172, 175]

[5] T.H. Clausen *et al.*, "The optimized link state routing protocol version 2," February 2007, Internet Draft, `http://www.ietf.org/internet-drafts/draft-ietf-manet-olsrv2-03.txt` (work in progress). [23]

[6] D. Johnson *et al.*, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," Request for Comments, `http://www.ietf.org/rfc/rfc4728.txt`. [24]

[7] C. Perkins *et al.*, "Ad hoc on-demand distance vector (aodv) routing," Request for Comments, `http://www.ietf.org/rfc/rfc3561.txt`. [24, 99]

[8] I. Chakeres *et al.*, "Dynamic manet on-demand (dymo) routing," February 2007, Internet Draft, `http://www.ietf.org/internet-drafts/draft-ietf-manet-dymo-08.txt` (work in progress). [24, 99]

[9] L. Kleinrock H. Takagi, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 246–257, March 1984. [24]

[10] I. Stojmenović H. Frey, "Geographic and energy aware routing in sensor networks," in *Handbook of Sensor Networks: Algorithms and Architectures*, I. Stojmenović, Ed., chapter 12, pp. 381–415. Wiley, 2006. [24, 69]

[11] B. Karp *et al.*, "Greedy perimeter stateless routing (gpsr)," `http://www.icir.org/bkarp/gpsr/gpsr.html`. [24, 167]

[12] Young-Bae Ko and Nitin H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," in *Proc. of the 4th annual ACM/IEEE international conference on Mobile computing and networking (MOBICOM'98)*, 1998, pp. 66–75. [25]

[13] Z.J. Haas *et al.*, "Zone routing protocol (zrp)," July 2002, Internet Draft, `http://tools.ietf.org/id/draft-ietf-manet-zone-zrp-04.txt`(work in progress). [25]

[14] N. Nikaein, C. Bonnet, and N. Nikaein, "Hybrid ad hoc routing protocol (harp)," in *Proc. of the International Symposium on Telecommunications (IST'01)*, 2001. [25]

[15] W. Creixell and K. Sezaki, "Mobility prediction algorithm for mobile ad hoc network using pedestrian trajectory data," in *Proc. of the IEEE International Region 10 Conference (TENCON'04)*, November 2004. [30, 31]

[16] Z. Zaidi and B. Mark, "Mobility estimation based on an autoregressive model," 2004, Submitted to IEEE Transactions on Vehicular Technology, Jan. 2004. (Pre-print) Available at URL: `http://mason.gmu.edu/zzaidi`. [30, 31]

[17] T. Liu, P. Bahl, and I. Chlamtac, "Mobility modeling location tracking and trajectory prediction in wireless atm networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 6, pp. 922–936, 1998. [30, 32, 33, 35]

[18] P.N. Pathirana, A.V. Savkins, and S. Jha, "Robust extended kalman filter based technique for location management in pcs networks," *Elsevier Computer Communications*, vol. 27, pp. 502–512, 2004. [30, 33]

[19] I. Guvenc *et al.*, "Enhancement to rss based indoor tracking systems using kalman filters," in *Proc. of the Global Signal Processing Conf. (GSPx)*, April 2003. [30]

[20] Z. Zaidi and B. L. Mark, "Real-time tracking algorithms for cellular networks based on kalman filtering," *IEEE Transactions on Mobile Computing*, vol. 4, no. 2, pp. 195–208, March-April 2005. [30, 33, 35]

[21] S-Z Yu and H. Kobayashi, "An integrated mobility and traffic model for resource allocation in wireless networks," in *3rd ACM Workshop on Wireless Mobile Multimedia*, 2000, pp. 39–47. [30, 32]

[22] S-Z Yu and H. Kobayashi, "A hidden semi-markov model with missing data and multiple observation sequences for mobility tracking," *ACM Signal Processing*, vol. 83, no. 2, pp. 235–250, February 2003. [30, 33]

[23] B. L. Mark and Z. Zaidi, "Robust mobility tracking for cellular networks," in *Proc. of IEEE International Conference on Communications (ICC'02)*, May 2002. [30, 33]

[24] Z. Yang and X. Wang, "Joint mobility tracking and hard hand-off in cellular networks via sequential mote carlo filtering," in *Proc. of the IEEE INFOCOM*, June 2002. [30, 35]

[25] Fredrik Gustafsson *et al.*, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, February 2002. [30, 35]

[26] L. Mihaylova *et al.*, "Mobility tracking in cellular networks with sequential monte carlo filters," in *Proc. of the IEEE International Conference on Information Fusion*, July 2005. [30, 35]

[27] Z. Shah and A. Malaney, "Particle filters and position tracking in wi-fi networks," in *Proc. of the 63rd Vehicular Technology Conference (VTC'06 spring)*, 2006, vol. 2, pp. 613–617. [30, 35]

[28] B. Liang and Z.J Haas, "Predictive distance-based mobility management for pcs networks," *IEEE Transactions on Networking*, vol. 11, no. 5, pp. 1–15, October 2005. [31]

[29] J. Krumm, "Probabilistic inferencing for location," in *Proc. of Location-Aware Computing*, October 2003. [31]

[30] R.A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Transactions Aerospace and Electronic Systems*, vol. 6, pp. 473–483, July 1970. [32]

[31] G. Liu and G. Maguire, "A class of mobile motion prediction algorithms for wireless mobile computing," *Mobile Networks and Applications (MONET)*, vol. 1, no. 2, pp. 113–121, 1996. [35]

[32] *A regular path recognition method and prediction of user movements in wireless networks*, 2001. [35]

[33] J-M. François and Guy Leduc, "An entropy-based knowledge spreading and application to mobility prediction," in *Proc. of the 1st ACM/e-NEXT International Conference on Future Networking Technologies (CoNext'05)*, 2005. [35]

[34] A. Bhattacharya and S.K. Das, "Lezi-update: An information-theoretic approach to track moble users in pcs networks," *ACM Wireless Networks (WINET)*, vol. 8, no. 2-3, pp. 121–135, 2002. [37]

[35] F. Yu and V. Leung, "Mobility-based call admission control and bandwidth reservation in wireless cellular networks," *Elsevier Computer Networks*, vol. 38, pp. 577–589, 2001. [37]

[36] J. Gil *et al.*, "Restoration scheme of mobility databases by mobility learning and prediction in pcs networks," *IEEE Journal on Selected Area in Communications (JSAC)*, vol. 19, no. 10, pp. 1962–1973, 2001. [38]

[37] R. Chellappa, A. Jennings, and N. Shenoy, "The sectorized mobility prediction algorithm for wireless networks," in *Proc. of the Int'l Conference on Information and Communication Technologies*, 2003. [39]

[38] D. Levine, I. Akyildiz, and M Naghshineh, "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 1–12, February 1997. [40]

[39] I.F. Akyildiz and W. Wang, "The predictive user mobility profile framework for wireless multimedia networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 1021–1035, December 2004. [40]

[40] J. Capka and R. Boutaba, *Mobility Prediction in Wireless Networks*, vol. 3271 of *Lecture Notes in Computer Science*, pp. 320–333, Springer, 2004. [40]

[41] Y. Shang, W. Guo, and S. Cheng, *Clustering Algorithm Based on Wavelet Neural Network Mobility Prediction in Mobile Ad Hoc Network*, vol. 3498 of *Lecture Notes in Computer Science*, pp. 391–396, Springer, 2005. [41]

[42] J. Haerri, N. Nikaein, and C. Bonnet, "Trajectory knowledge for improving topology control in mobile ad-hoc networks," in *Proc. of the 1st ACM/e-NEXT International Conference on Future Networking Technologies (CoNext'05)*, 2005. [41]

[43] J. Haerri, F. Filali, and C. Bonnet, *On the application of mobility predictions to multipoint relaying in MANETs: kinetic multipoint relays*, vol. 3837 of *Lecture Notes in Computer Science*, pp. 143–155, Springer, 2005. [41]

[44] V. Kumar and S.R. Das, "Performance of dead-reckoning based location service for mobile ad hoc networks," *ACM Wireless Communications and Mobile Computing Journal*, vol. 4, no. 2, pp. 189–202, March 2004. [42]

[45] S. Sharma *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. [42]

[46] X. Luo, T. Camp, and 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. [42]

[47] T. King *et al.*, "Dead-reckoning for position-based forwarding on highways," in *Proc. of the 3rd International Workshop on Intelligent Transportation (WIT 2006)*, 2006. [42]

[48] C. Doss, R. A. Jennings, and 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. [42]

[49] S. Jiang and D. He, "A prediction-based link availability estimation for routing metrics in manets," *IEEE/ACM Transaction on Networking*, vol. 13, no. 6, pp. 1302–1312, 2005. [42]

[50] M. Musolesi, S. Hailes, and 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. [42]

[51] W. Su, S. Lee, and M. Gerla, "Mobility prediction in wireless networks," in *IEEE Military Communication Conference (MILCOM)*, 2000, vol. 1, pp. 491–495. [43]

[52] A. Agarwal and S. R. Das, "Dead reckoning for mobile ad hoc networks," in *Proc. of the 2003 IEEE Wireless Communications and Networking Conference (WCNC'03)*, 2003. [43]

[53] N.C. Wang and S.W. Chang, "A reliable on-demand routing protocol for mobile ad hoc networks with mobility prediction," *Elsevier Computer Communications*, vol. 29, pp. 123–135, 2005. [43]

[54] H. Menouar, M. Lenardi, and 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. [43]

[55] J. Haerri, F. Filali, and C. Bonnet, "Kinetic link state routing," Technical Report 07-196, Institut Eurécom, 2007. [43]

[56] Marco Fiore, "Mobility models in inter-vehicle communications literature," Technical report, Politecnico di Torino, 2006. [46, 56]

[57] A. Rojas, P. Branch, and G. Armitage, "Experimental validation of the random waypoint mobility model through a real world mobility trace for large geographical areas," in *Proc. of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2005. [46]

[58] W. Hsu *et al.* , "Weighted waypoint mobility model and its impact on ad hoc networks," *ACM Mobile Computer Communications Review (MC2R)*, vol. 9, no. 1, pp. 59–63, January 2005. [46]

[59] Evaluating The HWGui Project Visualizing and Transforming Movement Patterns of Vehicles on Highways, ," `http://www.informatik.uni-mannheim.de/pi4.data/content/projects/hwgui`. [47]

[60] Mirco Musolesi and Cecilia Mascolo, "A community based mobility model for ad hoc network research," in *Proc. of the 2nd ACM/SIGMOBILE International Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN'06)*, 2006. [47]

[61] UDel Models For Simulation of Urban Mobile Wireless Networks, ," `http://udelmodels.eecis.udel.edu/`. [47, 65]

[62] Qunwei Zheng, Xiaoyan Hong, and Jun Liu, "An agenda-based mobility model," in *Proc. of the 39th IEEE Annual Simulation Symposium (ANSS-39-2006)*, 2006. [47]

[63] Community Resource for Archiving Wireless Data At Dartmouth (Crawdad), ," `http://crawdad.cs.dartmouth.edu`. [48]

[64] Realistic Vehicular Traces, ," `http://lst.inf.ethz.ch/ad-hoc/car-traces`. [48]

[65] UMass Diverse Outdoor Mobile Environment (DOME), ," `http://prisms.cs.umass.edu/dome`. [48]

[66] MIT Media Lab: Reality Mining, ," `http://reality.media.mit.edu`. [48]

[67] Community wide Library of Mobility and Wireless Networks Measurements, ," `http://nile.usc.edu/MobiLib`. [48]

[68] Cristian Tuduce and Thomas Gross, "A mobility model based on wlan traces and its validation," in *Proc. of the IEEE INFOCOM*, 2005. [48]

[69] Jungkeun Yoon *et al.*, "Building realistic mobility models from coarse-grained traces," in *Proc. of the ACM International Conference On Mobile Systems, Applications And Services (MobiSys'06)*, 2006. [48]

[70] Minkyong Kim, David Kotz, and Songkuk Kim, "Extracting a mobility model from real user traces," in *Proc. of the 26th Annual IEEE INFOCOM*, 2006. [48]

[71] J. Hou *et al.*, "Modeling steady-state and transient behaviors of user mobility: Formulation, analysis, and application," in *Proc. of the the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'06)*, 2006. [48]

[72] Augustin Chaintreau *et al.*, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *Proc. of the 26th Annual IEEE INFOCOM*, 2006. [48, 49]

[73] V. Srinivasan *et al.*, "Analysis and implications of student contact patterns derived from campus schedules," in *Proc. of the The 12th ACM International Conference on Mobile Computing and Networking (MOBICOM'06)*, 2006. [48]

[74] Paramics: Microscopic Traffic Simulation, ," `http://www.paramics-online.com/`. [49, 58]

[75] CORSIM: Microscopic Traffic Simulation Model, ," `http://www-mctrans.ce.ufl.edu/featured/TSIS/Version5/corsim.htm`. [49, 58, 134]

[76] ptv simulation VISSIM, ," `http://www.english.ptv.de/cgi-bin/traffic/traf_vissim.pl`. [49, 58]

[77] TRANSIMS, ," `transims.tsasa.lanl.gov`. [49, 58]

[78] VanetMobiSim Project Home Page, ," `http://vanet.eurecom.fr`. [50, 59, 60, 64, 115]

[79] L. Bononi *et al.*, "Parallel and distributed simulation of wireless vehicular ad hoc networks," in *Proc. of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2006. [50, 59, 60, 65]

[80] C. Gorgorin *et al.*, "An integrated vehicular and network simulator for vehicular ad-hoc networks," in *Proc. of the European Simulation and Modelling Conference (ESM)*, 2006. [50, 59, 60, 65]

[81] F. Karnadi, Z. Mo, and K.-C. Lan, "Rapid generation of realistic mobility models for vanet," in *Proc. of the IEEE Wireless Communication and Networking Conference (WCNC)*, 2007. [51, 59, 60, 62]

[82] SUMO Simulation of Urban MObility, ," `http://sumo.sourceforge.net`. [51, 59, 60, 61]

[83] H. Wu *et al.*, "Simulated vehicle-to-vehicle message propagation efficiency on atlanta's i-75 corridor," in *Proc. of the Transportation Research Board Conference*, 2005. [51]

[84] The Federated Simulations Development Kit (FDK), ," `http://www-static.cc.gatech.edu/computing/pads/fdk.html`. [52, 58]

[85] Andrew Chen *et al.*, "Impact of transmission power on tcp performance in vehicular ad hoc networks," in *Proc. of the 4th IEEE/IFIP Wireless On demand Networks and Services*, 2007. [52, 59, 60]

[86] Rama Vuyyuru and Kentaro Oguchi, "Vehicle-to-vehicle ad hoc communication protocol evaluation using simulation framework," in *Proc. of the 4th IEEE/IFIP Wireless On demand Networks and Services*, 2007. [52]

[87] Traffic and Network Simulation Environment, ," `http://wiki.epfl.ch/trans/`. [52, 59, 60, 62]

[88] Multiple Simulator Interlinking Environment for C2CC in VANETs, ," `http://www.cn.uni-duesseldorf.de/projects/MSIE`. [52]

[89] C. Schroth *et al.*, "Simulating the traffic effects of vehicle-to-vehicle messaging systems," in *Proc. of the 5th International Conference on ITS Telecommunications*, 2005. [52, 59, 60, 62]

[90] M. Brackstone and M. McDonald, "Car-following: a historical review," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 4, pp. 181–196, 1999. [56]

[91] Sakda Panwai and Hussein Dia, "Comparative evaluation of microscopic car following behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 314–325, 2005. [56]

[92] Christian Gawron Stefan Krauss, Peter Wagner, "Metastable states in a microscopic model of traffic flow," *Physical Review E*, vol. 55, no. 5, pp. 55–97, 1997. [56]

[93] Kai Nagel and Michael Schreckenberg, "A cellular automaton model for freeway traffic," *Journal de Physic I*, vol. 2, pp. 2221–2229, 1992. [56]

[94] R. Wiedemann, "Simulation des straenverkehrsflusses," Schriftenreihe heft 8, institute for transportation science, University of Karlsruhe, 1974. [56]

[95] Robert E. Chandler *et al.*, "Traffic dynamics: Studies in car following," *Operations Research*, vol. 6, no. 2, pp. 165–184, 1958. [56]

[96] P. G. Gipps, "A behavioral car following model for computer simulation," *Transportation Research Board*, vol. 15, pp. 105–111, 1981. [56]

[97] M. Trieber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000. [56, 122]

[98] P. G. Gipps, "A model for the structure of lane changing decisions," *Transportation Research Board*, vol. 20, pp. 107–120, 1986. [57]

[99] D. Chowdhury, D.E. Wolf, and M. Schreckenberg, "Particle hopping models for two-lane traffic with two kinds of vehicles:effects of lane-changing rules," *Physica A Statistical Mechanics and its Applications*, vol. 235, pp. 417–439, February 1997. [57]

[100] K.I. Ahmed, *Modeling Drivers' Acceleration and Lane Changing Behavior*, Ph.D. thesis, Massachusetts Institute of Technology (MIT), 1999. [57]

[101] M. Treiber and D. Helbing, "Realistische mikrosimulation von strassenverkehr mit einem einfachen modell," in *16th Symposium Simulationstechnik (ASIM'02)*, 2002. [57, 125]

[102] R. Wiedemann, "Modeling of rti-elements on multi-lane roads," in *Advanced Telematics in Road Transport*, vol. DG XIII. Commission of the European Community, 1991. [57]

[103] M. Fiore, J. Harri, F. Filali, and C. Bonnet, "Vehicular mobility simulation for vanets," in *Proc of the 40th IEEE/SCS Annual Simulation Symposium (ANSS'07)*, 2007. [57]

[104] B. Zhou, K. Xu, and M. Gerla, "Group and swarm mobility models for ad hoc network scenarios using virtual tracks," in *Proc. of IEEE Military Communications Conference (MILCOM)*, 2004. [59, 60]

[105] U.S. Census Bureau Topologically Integrated Geographic Encoding and Referencing (TIGER) system, ," `http://www.census.gov/geo/www/tiger`. [59, 61, 64, 75, 106, 120]

[106] F. Bai, N. Sadagopan, and A. Helmy, "The important framework for analyzing the impact of mobility on performance of routing for ad hoc networks," *AdHoc Networks Journal*, vol. 1, no. 4, pp. 383–403, 2006. [59, 60, 61]

[107] BonnMotion, ," `http://web.informatik.uni-bonn.de/IV/BonnMotion`. [59, 60, 61]

[108] A. K. Saha and D. Johnson, "Modeling mobility for vehicular ad hoc networks," in *1st ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004)*, 2004, Poster Session. [59, 60, 61]

[109] ESRI Shape Files, ," `http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf`. [59, 62]

[110] The SHIFT Traffic Simulator, ," `http://path.berkeley.edu`. [59, 60, 62]

[111] D. Choffnes and F. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks," in *2nd ACM Workshop on Vehicular Ad Hoc Networks (VANET 2005)*, 2005. [59, 60, 62]

[112] et *al.* R. Mangharam, "Groovesim: a topography-accurate simulator for geographic routing in vehicular networks," in *2nd ACM Workshop on Vehicular Ad Hoc Networks (VANET 2005)*, 2005. [59, 60, 62]

[113] The Obstacle Mobility Model, ," `moment.cs.ucsb.edu/mobility/`. [59, 60, 61]

[114] H. M. Zimmermann and I. Gruber, "A voronoi-based mobility model for urban environments," in *Proc. of the European Wireless 2005 (EW'05)*, 2005. [59, 60, 64]

[115] M.J. Feeley, N.C. Hutchinson, and S. Ray, *Realistic Mobility for Mobile Ad Hoc Network Simulation*, vol. 3158 of *Lecture Notes in Computer Science*, pp. 324–329, Springer, 2004. [59, 60, 61]

[116] CANU Project Home Page, ," `http://canu.informatik.uni-stuttgart.de`. [59, 60, 63, 115, 119, 122]

[117] Ertico, ," `http://www.ertico.com/en/links/links/gdf_-_geographic_data_files.htm`. [59, 64, 120]

[118] S. Jaap, M. Bechler, and L. Wolf, "Evaluation of routing protocols for vehicular ad hoc networks in city traffic scenarios," in *Proc. of the 5th International Conference on Intelligent Transportation Systems Telecommunications (ITST)*, 2005. [59, 60, 63, 144, 146]

[119] K. Konishi *et. al*, "Mobireal simulator – evaluating manet applications in real environments," in *Proc. of 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS2005)*, 2005. [59, 60, 64]

[120] A. Mahajan *et al.*, "Evaluation of mobility models for vehicular ad-hoc network simulations," Technical Report 051220, Florida State University, 2005. [59, 60, 63]

[121] GPS TrackMaker, ," `http://www.gpstm.com`. [59, 65]

[122] Jijun Yin *et al.*, "Performance evaluation of safety applications over dsrc vehicular ad hoc networks," in *Proc. of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, 2004. [58]

[123] Hao Wu *et al.*, "Mddv: Mobility-centric data dissemination algorithm for vehicular networks," in *Proc. of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, 2004. [58]

[124] S. Boxill and L. Yu, "An evaluation of traffic simulation models for supporting its development," Technical Report 167602-1, Texas Southern University, 2000. [58]

[125] Dirk Helbing, "Traffic and related self-driven many-particle systems," *Rev. Mod. Phys.*, vol. 73, no. 4, pp. 1067–1141, December 2001. [61]

[126] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad noc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002. [61]

[127] F. Bai and A. Helmy, "A survey of mobility modeling and analysis in wireless ad hoc networks," available at `http://nile.usc.edu/important/chapter1.pdf`, 2004. [61]

[128] J.Y. Le Boudec and M. Vojnovic, "The Random Trip Model: Stability, Stationary Regime, and Perfect Simulation," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1153–1166, 2006. [61, 102, 128]

[129] J. Haerri, F. Filali, and C. Bonnet, "A framework for mobility models generation and its application to inter-vehicular networks," in *Proc. of the 3rd IEEE International Workshop on Mobility Management and Wireless Access (MobiWac'05)*, 2005. [61]

[130] The Network Simulator (ns 2), ," `http://www.isi.edu/nsnam/ns`. [61, 119]

[131] Global Mobile Information Systems Simulation Library, ," `http://pcl.cs.ucla.edu/projects/glomosim`. [61, 119]

[132] QualNet Developer, ," `http://www.scalable-networks.com`. [61, 119]

[133] The OPNET Modeler, ," `http://www.opnet.com/products/modeler/home.html`. [61, 119]

[134] Radio Propagation Modeling, ," `http://www.ipvs.uni-stuttgart.de/abteilungen/vs/abteilung/mitarbeiter/eigenes/stepania/start/downloads`. [63]

[135] ARTIS: Advanced Runtime Infrastructure, ," `http://pads.cs.unibo.it/dokuwiki/doku.php?id=pads:artis-doc`. [65]

[136] J. Basch *et al.*, "Data structures for mobile data," *Journal of Algorithms*, vol. 31, no. 1, pp. 1–28, 1999. [68]

[137] S. Bespamyatnikh *et al.*, "Mobile facility location," in *Proc. of the 4th international Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2000. [68]

[138] J. Herschberger, "Smooth kinetic maintenance of clusters," in *Proc. of the 19th Annual Symposium on Computational Geometry*, 2003. [68, 69]

[139] L. Guibas, "Kinetic data structures: A state of the art report," in *Proc of the 3rd Workshop of Algorithmic Foundations of Robotics*, 1998. [68]

[140] X-Y. Li *et al.*, "Localized delaunay triangulation with application in wireless ad hoc networks," *IEEE Trans on Parallel and Distributed Processing*, vol. 14, no. 10, pp. 1035–1047, 2003. [69]

[141] K. Alzoubi *et al.*, "Geometric spanners for wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Processing*, vol. 14, no. 4, pp. 408–421, 2003. [69]

[142] N. Li, J.C. Hou, and L. Sha, "Design and analysis of an mst-based topology control algorithm," in *Proc. of the IEEE INFOCOM*, 2003. [69, 86, 88, 89, 175]

[143] N. Li and J.C. Hou, "Blmst: A scalable, power-efficient broadcast algorithm for wireless networks," in *Proc. of the 1st International Conference on Quality of Service in Heterogenous Wired/Wireless Networks (QSHINE'04)*, 2004. [69]

[144] J. Cartigny, F. Ingelrest, and D.Simplot, "Lmst and rng based minimum-energy broadcast protocols in ad hoc networks," *Ad Hoc Networks*, vol. 3, no. 1, pp. 1–16, 2005. [69]

[145] X.Y. Li and I. Stojmenović, "Broadcasting and topology control in wireless ad hoc networks," in *Handbook of Algorithms for Mobile and Wireless Networking and Computing*, A. Boukerche and I. Chlamtac, Eds., chapter 11, pp. 239–264. CRC Press, 2006. [69, 79, 170, 174]

[146] R.J. Fontana and S.J. Gunderson, "Ultra-wideband precision asset location system," in *Proc. of the IEEE Conference on Ultra Wideband Systems and Technologies*, 2002. [70]

[147] C. Gentile and L. Klein-Berndt, "Robust location using system dynamics and motion constraints," in *Proc. of the IEEE International Conference on Communications (ICC)*, 2004. [70]

[148] ," Wikipedia - World Geodetic System, `http://en.wikipedia.org/wiki/WGS84`. [71]

[149] J. Härri *et al.*, "Manet position and mobility signaling format," February 2007, Internet Draft, `http://tools.ietf.org/id/draft-haerri-manet-position-signaling-00.txt` (work in progress). [79]

[150] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transaction on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000. [87]

[151] S. Narayanaswamy *et al.*, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol," in *Proc. of the European Wireless*, 2002. [87, 88]

[152] L. Li *et al.*, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proc. ACM Symposium on Principles of Distributed Computing*, 2001. [87, 88]

[153] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. of IEEE INFOCOM*, 2000. [87]

[154] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 17, no. 8, pp. 1333–1344, 1999. [87]

[155] Xiang-Yang Li *et al.*, "Applications of k-local mst for topology control and broadcasting in wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 12, pp. 1057–1069, 2004. [87, 88]

[156] F.J. Ovalle-Martinez *et al.*, "Finding minimum transmission radii and constructing minimal spanning trees in ad hoc and sensor networks," *Journal of Parallel and Distributed Computing*, vol. 65, no. 2, pp. 132–141, 2005. [87, 88]

[157] Aaron Zollinger Roger Wattenhofer, "Xtc: A practical topology control algorithm for ad-hoc networks," in *4th IEEE Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2004. [87, 88]

[158] Li Li and Joseph Y. Helpern, "A minimum-energy path-preserving topology-control algorithm," *IEEE Transactions on Wireless Communications*, vol. 3, no. 3, pp. 910–921, 2004. [87]

[159] P. Santi, D. M. Blough, and F. Vainstein, "A probabilistic analysis for the range assignment problem in ad hoc networks," in *Proc. ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2000. [88]

[160] S. Basagni, D. Turgut, and S. K. Das, "Mobility-adaptive protocols for managing large ad hoc networks," in *Proc. IEEE International Conference on Communications (ICC)*, 2001. [88]

[161] N. Nikaein, H. Labiod, and C. Bonnet, "Distributed dynamic routing algorithm for mobile ad-hoc networks," in *The ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2000. [88, 89, 90, 91]

[162] Ravi Prakash *et al.*, "A lower bound for broadcasting in mobile ad hoc networks," Technical Report IC/2004/37, Ecole Polytechnique Fédérale de Lausanne (EPFL), 2004. [98]

[163] F. Baccelli and P. Bremaud, *Palm Probabilities and Stationary Queues*, Springer LNS, 1987. [102, 103]

[164] Random Trip Framework, ," `http://lrcwww.epfl.ch/RandomTrip/`. [102, 104, 152, 153, 177]

[165] J-Y. Le Boudec, "Understand the simulation of mobility models with palm calculus," Technical Report EPFL/IC/2004/53, Ecole Polytechnique Fédérale de Lausanne (EPFL), 2004. [103, 104]

[166] B. Ghosh, "Random distances within a rectangle and between two rectangles," *Bulletin Calcutta Mathematical Society*, vol. 43, pp. 17–24, 1951. [103]

[167] William Navidi and Tracy Camp, "Stationary distributions for the random waypoint model," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004. [104]

[168] A. Jardosh *et al.*, "Toward realistic mobility models for mobile ad hoc networks," in *Proc. of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom 2003)*, 2003. [120]

[169] J. Tian *et al.*, "Graphbased mobility model for mobile ad-hoc network simulation," in *Proc. of the IEEE Annual Simulation Symposium (ANSS'02)*, 2002. [122]

[170] C. Bettstetter, "Smooth is better than sharp: A random mobility model for simulation of wireless networks," in *4th ACM International Work-shop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM'01)*, 2001. [122]

[171] I. Seskar *et al.*, "Rate of location area updates in cellular systems," in *Proc. of the IEEE Vehicular Technology Conference (VTC'92)*, 1992. [122]

[172] J. Broch *et al.*, "A performance comparison of multihop wireless ad hoc networking protocols," in *Proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Networking (MobiCom'98)*, 1998. [144, 145]

[173] S.R. Das, C.E. Perkins, and E.M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *Proc. of the IEEE INFOCOM*, 2000. [144, 145]

[174] Sven Jaap, Marc Bechler, and Lars Wolf, "Evaluation of routing protocols for vehicular ad hoc networks in typical road traffic scenarios," in *Proc. of the 11th EUNICE Open European Summer School on Networked Applications*, 2005. [144, 146]

[175] P. Johansson *et al.*, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *Proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Networking (Mobicom'99)*, 1999. [145]

[176] T. Clausen, P. Jacquet, , and L. Viennot, "Comparative study of routing protocols for mobile ad hoc networks," in *Proc. of the 1st IFIP Annual Mediterranean Ad Hoc Networking Workshop (MedHocNet'02)*, 2002. [146]

[177] The FleetNet Project, ," `http://www.et2.tu-harburg.de/fleetnet/english/vision.html`. [146]

[178] R.A. Santos *et al.*, "Performance evaluation of routing protocols in vehicular ad hoc networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 1, no. 1, pp. 80–91, 2005. [146]

[179] The Network on Wheels Project, ," `http://www.informatik.uni-mannheim.de/pi4/lib/projects/now/`. [146]

[180] M. Chiang and G. Carlsson, "Lora: Robust and simple routing algorithms for ad hoc mobile wireless networks," in *Proc. of the IEEE GlobeCom'01*, 2001, vol. 5, pp. 2793–2797. [167]

[181] Brad Williams and Tracy Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proc. ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002. [170, 174]

[182] A. Laouiti *et al.*, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," in *Proc. of the 35th Annual Hawaii International Conference on System Sciences (HICSS'2001)*, 2001. [170, 175]

[183] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 18, no. 3, pp. 535–547, 2000. [170, 175]

[184] C.-K Toh, H. Cobb, and D.A. Scott, "Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks," in *Proc. of the IEEE International Conference on Communications (ICC)*, 2001. [170, 175]

[185] A. Busson, N.Mitton, and E. Fleury, "An analysis of the multi-point relays selection in olsr," Technical Report 4568, INRIA Rhône-Aples, 2005. [171]

[186] Naval Research Lab OLSR (NRLOLSR), ," http://pf.itd.nrl.navy.mil/projects.php?name=olsr. [173, 177, 184]

[187] I. Stojmenović H. Frey, "Connected dominating set in sensor networks and manets," in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. M. Pardalos, Eds., pp. 329–369. Springer US, 2005. [175]

[188] O. Liang, Y. A. Sekercioglu, and N. Mani, "A survey of multipoint relay based broadcast schemes in wireless ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 30–46, 2006. [175, 191]

[189] J.P. Macker and Justin Dean, "Simplified multicast forwarding in mobile ad hoc networks," in *Proc. of the IEEE Military Communications Conference (MILCOM)*, 2004. [182]

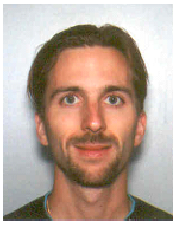
[190] Philippe Jacquet, "Performance of connected dominating set in olsr protocol," Technical Report 5098, INRIA Rocquencourt, 2004. [182]

[191] Brian Adamson and Justin Dean, "Simplified multicast forwarding (smf) update," in *Proc. of the 64th IETF Meeting*, 2005. [182]

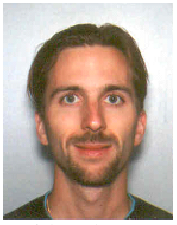
List of Tables

List of Figures

Curriculum Vitae



Jérôme Härri graduated from the Swiss Federal Institute (EPFL), Lausanne, Switzerland, in 2002, where he majored in Mobile Telecommunication Systems. He later obtained a Master degree in Signal Processing and Digital Communication from the University of Nice (UNSA),France. From March 2002 to April 2003, he worked as a Guest Researcher at the National Institute of Standards and Technology (NIST), Gaithersburg, MD USA, on Position Awareness benefits for Proactive Mobile Ad-Hoc routing protocols. In September 2003, he joined l'Institut Eurécom to pursue his PhD thesis under the supervision of Prof. Christian Bonnet and Prof. Fethi Filali. In 2006, he has been a visiting researcher at Prof. Mario Gerla's Network Research Lab at the University of California in Los Angeles (UCLA), for a period of 4 months. He is focusing on modeling and predicting mobility in wireless ad hoc networks. His other fields of interest are Localization, Distributed Computing, Security, Information Theory.



Jérôme Härri est ingénieur diplômé de l'Ecole Polytechnique Fédérale de Lausanne (EPFL), Suisse, en Systèmes de Communication, avec spécialisation en Communications Mobiles, promotion 2002. En parallèle, il a effectué un Cycle Postgrade à l'Université de Nice, en France, et a obtenu un Master en Traitement du Signal et Communications Numériques. De mars 2002 à avril 2003, il a travaillé au "National Institute of Standards and Technology" (NIST) à Gaithersburg, MD USA, sur les conséquences et avantages de l'utilisation de la géolocalisation sur des protocoles distribués sans fil basés sur des tables de routage. En septembre 2003, il a rejoint l'Institut Eurécom afin d'y effectuer une thèse de doctorat sous la supervision du Prof. Christian Bonnet et Prof. Fethi Filali dans le domaine de la modélisation et la prédiction de mobilité dans les réseaux mobiles ad-hoc sans fil, notamment les reseaux vehiculaires. En 2006, il a sejourné dans le laboratoire de recherche en réseaux du Prof. Mario Gerla à l'université de Californie de Los Angeles pendant 4 mois. Ses domaines de recherche sont les nouvelles générations de protocoles de routage basés sur la géolocalisation dans des réseaux mobiles distribués sans fil. Ces autres domaines de recherche sont la localisation, la sécurité ainsi que la théorie de l'information.

List of Publications

Book Chapters

- Jérôme Härri, and Christian Bonnet, "La Sécurité dans les Réseaux Mobiles de Télécommunication", in *La Sécurité dans les Réseaux Sans Fil et Mobiles*, Vol. 2, pp. 131–179, Traité IC2–Réseaux et Télécoms, Hermès France.

- Jérôme Härri, and Christian Bonnet, "La Sécurité dans les Réseaux Mobiles de Télécommunication Mobiles de Nouvelle Génération", in *La Sécurité dans les Réseaux Sans Fil et Mobiles*, Vol. 3, pp. 21–46, Traité IC2–Réseaux et Télécoms, Hermès France.

International Journal Publications

- J. Härri, F. Filali, C. Bonnet, Kinetic Mobility Management For Vehicular Ad Hoc Network Protocols, *submitted to Elsevier Computer Communications, Special Issue on Mobility Protocols for ITS/VANET*, July 1st, 2007.

- J. Härri, M. Fiore, F. Filali, C. Bonnet, "VanetMobiSim: A Configurable Traffic Model for Generating Realistic Mobility Patterns for VANETs", *submitted to Elsevier Ad Hoc Networks*, July 1st, 2007.

- J. Härri, F. Filali, C. Bonnet, "Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy", *IEEE Communications Surveys and Tutorials*, June 2007.

International Conference Publications

- M. Fiore, J. Härri, F. Filali, C. Bonnet, "Understanding Vehicular Mobility in Networking", *submitted to the 1st International Workshop on Mobile Vehicular Networks (MoVeNet'07).*

- J. Härri, F. Filali, C. Bonnet, "Kinetic Graphs: A Framework for Capturing the Dynamics of Mobile Structures in MANET", *submitted to the 10-th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWim'07).*

- J. Härri, F. Filali, C. Bonnet, "Rethinking the Overhead of Geo-localization Information for Vehicular Communications", *Proceedings of the 1st IEEE International Symposium on Wireless Vehicular Communications (IEEE WiVec'07).*, October 1 - October 3, 2007.

- M. Fiore, J. Härri, F. Filali, C. Bonnet, "Vehicular mobility simulation for VANETs", *Proceedings of the 40th IEEE Annual Simulation Symposium (ANSS-40 2007)*, March 25 - March 29, 2007, Norfolk, USA.

- J. Härri, M. Fiore, F. Filali, C. Bonnet, "VanetMobiSim: generating realistic mobility patterns for VANETs", *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET'06)*, September 29, 2006, Los Angeles, USA.

- J. Härri, B. Zhou, M. Gerla, F. Filali, C. Bonnet, "Neighborhood changing rate: an unifying parameter to characterize and evaluate data dissemination scenarios" *Proceedings of the 4th Annual Conference on Wireless On demand Network Systems and Services (IEEE/IFIP WONS'07)*, January 24-26, 2007, Obergurgl, Austria.

- J. Härri, F. Filali, C. Bonnet, "On meaningful parameters for routing in VANETs urban environments under realistic mobility patterns", *Proceedings of the 1st IEEE Workshop on Automotive Networking and Applications (AutoNet'06) (in conjunction with IEEE Globecom 2006)*, December 1, 2006 ? San Francisco, CA, USA.

- J. Härri, F. Filali, C. Bonnet, "Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns", *Proceedings of the 5th IFIP Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net'06)*, June 14-17, 2006, Lipari, Italy. **Invited Paper**

- J. Härri, F. Filali, C. Bonnet, "On the application of mobility predictions to multipoint relaying in MANETs: kinetic multipoint relays", *Proceedings of the 1st Asian Internet Engineering Conference (AINTEC'05)*, December 13 - 15, 2005, Bangkok, Thailand.

- J. Härri, F. Filali, C. Bonnet, "Kinetic multipoint relaying: improvements using mobility predictions", *Proceedings of the 7th International Working Conference on Active and Programmable Networks (IWAN'05,)*, November 21-23 2005, Sophia Antipolis, France.

- J. Härri, N. Nikaein, C. Bonnet, "Trajectory knowledge for improving topology control in mobile ad-hoc networks", *Proceedings of the 1st ACM/e-NEXT International Conference on Future Networking Technologies (Co-NEXT'05)*, 24-27 October, 2005, Toulouse, France.

- J. Härri, C. Bonnet, "A lower bound for vehicles' trajectory duration", *Proceedings of the 62nd IEEE Vehicular Technology Conference (VTC' Fall 2005)*, September 26-29, 2005, Dallas, USA. **Best technical Paper Award**.

- J. Härri, F. Filali, C. Bonnet, "OLSR and MPR: mutual dependences and performances", *Proceedings of the 4th IFIP Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net 2005)*, June 21-24, 2005, ?le de Porquerolles, France.

- J. Härri, F. Filali, C. Bonnet, "A framework for mobility models generation and its application to inter-vehicular networks" *Proceedings of the 3rd IEEE International Workshop on Mobility Management and Wireless Access (Mobiwac'05)*, June 13-16, 2005, Maui, Hawaii, USA.

IETF DRAFTS

- J. Härri, F. Filali, and C. Bonnet, "MANET Position and Mobility Signaling: Problem Statement", http://www.ietf.org/internet-drafts/draft-haerri-manet-position-problem-statement-01.txt, work in progress.
- J. Härri, F. Filali, and C. Bonnet, "MANET Position and Mobility Signaling Format", http://www.ietf.org/internet-drafts/draft-haerri-manet-position-signaling-01.txt, work in progress.
- J. Härri, F. Filali, and C. Bonnet, "MANET Generalized Location Signaling Format", http://www.ietf.org/internet-drafts/draft-haerri-manet-location-02.txt, work in progress.

INVITED TALKS

- J. Härri, M. Fiore "A realistic mobility simulator for vehicular ad hoc networks" *NEWCOM Workshop on Wireless Communications, in conjunction with IEEE ICC'06*, June 11, 2006, Istanbul, Turkey.
- Jérôme Härri, "VanetMobiSim: Generating Realistic Vehicular Mobility for Vanet Simulations", NoE Newcom/Hycon Workshop, Florence, Mai 2006.
- J. Härri, C. Bonnet, "Mobilité: subir ou prédire ?" *24ème Congrès DNAC (De Nouvelles Architectures pour les Communications)*, April 21-28, 2006, Split, Croatia.

POSTERS

- J. Härri, M. Fiore, F. Filali, C. Bonnet, "VanetMobiSim: a configurable simulator for generating realistic mobility patterns for VANETs", *NEWCOM Dissemination Day*, February 17, 2007, Paris, France.
- J. Härri, M. Fiore, F. Filali, C. Bonnet, "VanetMobiSim: Generating Realistic Mobility Patterns for VANETs", *3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET'06)*, September 29, 2006, Los Angeles, USA.

ORIGINAL TECHNICAL REPORTS

- J. Härri, F. Filali, and C. Bonnet, "Kinetic Localized Minimum Spanning Tree", Technical Report RR-07-197, Institut Eurécom, July 2007.

- J. Härri, F. Filali, and C. Bonnet, "Kinetic Link State Routing", Technical Report RR-07-196, Institut Eurécom, May 2007.
- J. Härri, and C. Bonnet, "Topology Management: Unifying Topology Control and CDS Management", Technical Report RR-06-179, Institut Eurécom, July 2007.
- J. Härri, F. Filali, and C. Bonnet, "The Challenges of Predicting Mobility", Technical Report RR-06-171, Institut Eurécom, August 2006.
- J. Härri, F. Filali, and C. Bonnet, "Analysis of vehicular mobility patterns on routing protocols", Technical Report RR-06-170, Institut Eurécom, June 2006.
- J. Härri, F. Filali, and C. Bonnet, "Performance Testing of OLSR using Mobility Predictions", Technical Report RR-06-157, Institut Eurécom, March 2006.
- J. Härri, C. Bonnet, "On the classification of routing protocols in mobile ad-hoc networks ", Technical Report RR-04-115, Institut Eurécom, August 2004.

Notes