Kinetic Graphs: A Framework for Capturing the Dynamics of Mobile Structures in MANET

Jérôme Härri
haerri@eurecom.fr

Christian Bonnet
bonnet@eurecom.fr

Fethi Filali
filali@eurecom.fr

Institut Eurécom
Department of Mobile Communications
B.P. 193
06904, Sophia Antipolis, France

Abstract
In Mobile Ad Hoc Networks (MANET), structures are built in order to improve network resource for broadcast or routing. Inspired by graph theory, most of those structures are built using fixed criteria, such as degree or distance, yet based only on local information. However, mobility alters the optimality of these localized structures, as the criteria dynamically varies with time. Since those criteria do not change homogeneously, a periodic maintenance wastes network resource, as it inefficiently acquires new values in an disorganized way. In this paper, we introduce the concept of Kinetic Graphs as a method to capture the dynamics of mobile structures and accordingly develop an efficient maintenance. Unlike the static counterpart, kinetic graphs are assumed to be continuously changing and edges are represented by time-varying weights. Kinetic graphs are a natural extension of static graphs and provide solutions to similar problems, such as convex hulls, spanning trees or connected dominating sets, but for continuously mobile networks.


General Terms: Algorithms, design, management, performance.

Keywords: Kinetic graph, mobility management, design methodologies, mobile structure, time varying weights, broadcast, MANET.

1. Introduction
Mobile Ad Hoc Networks (MANETs) are an emergent concept in view for infrastructure-less communications. 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. Most of the network solutions for broadcasting, routing, or managing the topology in MANETs have been inspired from graph theory. Yet, due to the limited capability of processing power, storage and energy supply, many conventional graph algorithms are too complicated to be implemented for wireless ad hoc networks. They in fact require efficient distributed and localized algorithms with low computation complexity and low communication complexity. The community therefore worked on adapting decades of graph theory outbreaks to distributed computing, with the clear objective to obtain similar results as the centralized approaches. A survey on Localized approaches for broadcasting and topology control may be found in [10], and in [5] for routing.

Despite the efficient distributed solutions that have been designed, a major assumption has been widely ignored: mobility. Advocates of distributed solutions argued that mobility could be simply solved by periodic maintenance duty cycles that are optimally kept local. Yet, this kind of maintenance generates a waste of resource, instability and delays. Besides, it only adapt structures to past topologies, as any information that has been used to build a structure is invalidated due to mobility. Introduced by Bash et al. [1] ten years ago, Kinetic Data Structures (KDS) have been specifically developed as a mean to efficiently adapt data structures to mobile objects and attributes. KDS assume that trajectories of objects are known, but not when they will change. This is a direct use of mobility predictions applied to data structures. A detailed survey on KDS can be found in [4]. However, efficient localized and distributed algorithms were not available at that time. Therefore, most graph algorithms used in KDS are centralized.

When looking at the “state of the art” achievements in the approaches described in the previous paragraphs, we can see a straightforward interweaving aspect. Mobility has been studied for centralized graph algorithms, while localized graph algorithms have been 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. To the best of our knowledge, only few works [3, 8] appeared to have considered the potential benefits from joining both worlds. Figure 1 illustrates the two separate, yet complementary, issues of graph algorithms: central vs. local and static vs. dynamic methods.

In this paper, we propose to specifically regroup those two research areas and introduce the Kinetic Graphs framework. It consists of a neighborhood discovery process, a trajectory modeling, a time varying link weight computation, and finally an aperiodic neighborhood maintenance. By following the framework, any localized ad hoc network protocol may be adapted to the kinetic approach, including Topology Control, Broadcasting or Routing. The
approach is independent of the network model, the criteria chosen to build the backbone, or the localized MANET protocol used, and various approaches or combinations may be tested.

In the rest of this paper, we describe the framework and illustrate how we manage to build and maintain mobile structures in MANETs without requiring to the periodic beaconing process widely used by almost all network protocols, yet guaranteeing at least a similar network performance.

2. KINETIC GRAPH FRAMEWORK

In this section, we describe the Kinetic Graph framework, which consists of four steps: (i) a representation of nodes’ trajectories, (ii) a common message format for the posting of those trajectories, (iii) a time varying link weight for building the kinetic graphs based on the nodes’ trajectories, (iv) an aperiodic neighborhood maintenance in order to acquire nodes stealthily entering the neighborhood.

2.1 Trajectory Modeling

The first step in the Kinetic Graph framework is the modeling of nodes’ trajectories. For that matter, various mobility prediction techniques described in [6] may be applied. Depending on the required prediction efficiency, simple models based on first or second order kinematic model may be used. For more sophisticated predictions, Kalman or particle filters could also be envisioned.

Due to space limitations, we cannot provide an extensive description of the process of predicting mobility. In short, the first step is to define kinematic hypothesis in order to reduce the complexity of the kinetic model. As an example and without loss of generality, we assumed a fixed velocity between two successive trajectories in this paper, and therefore used a first order prediction model. Although the use of such a simplistic kinematic model generates some prediction errors and might look unrealistic, the objective of this paper is only to show the principle behind the Kinetic Graphs. 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 any other GPS-free solution and exchanged by means of beacon messages during the neighbor discovery phase. 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.

According to the first order kinematic model, we assume that each such node, say \( i \), follows a linear trajectory. Its position as a function of time is then described by

\[
\text{Pos}_i(t) = \begin{bmatrix} x_i + dx_i \cdot t \\ y_i + dy_i \cdot t \end{bmatrix}.
\]

where \( \text{Pos}_i(t) \) 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 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

\[
D^2_{ij}(t) = (D^2_{ji}(t) = ||\text{Pos}_j(t) - \text{Pos}_i(t)||^2)
\]

\[
= \left( \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \cdot t \right)^2
\]

\[
= a_{ij}t^2 + b_{ij}t + c_{ij},
\]

where \( a_{ij} \geq 0, c_{ij} \geq 0 \). Consequently, \( a_{ij}, b_{ij}, c_{ij} \) are defined as the three parameters describing node \( j \)'s trajectory with respect to node \( i \).

2.2 Neighborhood Discovery

Basically, the Kinetic Graph neighborhood discovery procedure lets a node detect changes in its neighborhood without any exchange of periodical beacon messages. During this phase, each node broadcasts a single\(^2\) Hello message indicating its presence in the neighborhood, and transmitting its mobility parameters. Such message is emitted using maximum transmission power in order to reach the maximum number of neighbors. Moreover, it 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 according to their predictions, 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.

As mentioned in the previous section, Kinetic Graphs require geo-localization information, and that is precisely during the Neighborhood Discovery phase that nodes exchange such information. Due to the recent outbreak of this field, no common agreement has been reached neither on the transmission format nor on the content to be transmitted. Moreover, the cost of transmitting GPS data has also been widely ignored by the ITS community.

Due to space restriction, we cannot describe here our proposed common signaling format and compression schema for the transmission of geo-localization data. We refer the interested reader to [7], where a common message format for the transmission of geo-localization data is described, and where the overhead generated by its use in vehicular communications is addressed. We particularly introduced a compression method reaching up to 70% overhead reduction at no precision loss. This solution has also been proposed for a possible standardization within the IETF [2].

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

\(^1\)Unless otherwise specified, we are considered moving in a two-dimensional plane.

\(^2\)In order to take into account possible collision and packet losses, a Hello message is sent a configurable number of times. Unless otherwise specified, we send each Hello message 3 times.
dense, the overhead induced by the transmission of these geo-localization data increases significantly. Figure 2 illustrates the cost of the transmission of geo-localization data as a function of the node degree. We can see that transmitting geo-localization without compression becomes a serious limiting factor for efficient network usage, as each packet could reach more than 1kbytes for dense networks. When using the compression proposed in [7], we can significantly reduce this drawback, which in turn could help improve the network protocols in general and Kinetic Graphs in particular.

To illustrate the per packet overhead for geo-localization data transmission, Figure 2 shows the Neighbor Discovery Overhead in bytes/packet for different node densities. The overhead is measured for Node ID, Cartesian, GPS, Compressed Cartesian, and Compressed GPS. The figure clearly demonstrates the importance of using efficient compression techniques.

2.3 Time Varying Link Weights

In this section, we describe the most important step of the Kinetic Graph framework: the extension to mobile structures of the popular link weight concept widely used in graph theory. The time varying link weights, opposed to static link weights, are used to build and dynamically update an optimal graph without requiring a periodic resampling of each link weight. Most of graph algorithms may be adapted to use time varying link weights. However, as mentioned in the introduction part of this paper, it is important that graph protocols be distributed and local. Accordingly, we suggest potential targets for mobile graph constructions described in [10].

We first 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 kinetic graph theory, we have basically the same definitions as in graph theory, yet adapted to moving structures:

- **Time Varying Link Weight** – It is a continuous and integrable function relating the evolution of the link weight between two graph vertices 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 between two graph vertices 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, the trajectory modeling described in Section 2.1, and Figure 3, we are now ready to illustrate the difference between static and kinetic graphs. Considering the euclidian distance between two nodes as the link weight, and the shortest link weight as criterion, the optimal tree shown in Figure 3(a) is generated. This tree is yet only valid at the precise moment when it is built, as mobility makes link weights and thus the criteria change.

![Figure 2: Illustration of the per packet overhead for geo-localization data transmission](image)

![Figure 3: Example of the Kinetic Graph Approach](image)
Figure 4: Graphical representation of two time varying link weights and the computation of the transition and respective activations

as the time varying link weight used in this paper, the kinetic criterion is defined as follows:

\[ \text{Crit}_i(t_1) \rightarrow W_{ij}(t_1) \text{ iff } W_{ij}(t_1) = \min_{k \in n_b_i} (W_{ik}(t_1)) \]

Then, based on the time varying link weights and the kinetic criterion, the activation of a link between node \(i\) and node \(j\) over an interval \([t_1, t_2]\) is defined as

\[ \text{act}(i, j)[t_1, t_2] = \begin{cases} \min_{t_1} \text{s.th. } \text{Crit}_i(t_1) \rightarrow W_{ij}(t_1) \\ \max_{t_2 \in (t_1, \infty)} \text{s.th. } \text{Crit}_i(t_2) \rightarrow W_{ij}(t_2) \end{cases} \]

According to the defined time varying link weights and the required kinetic criterion, this procedure manages to compute the set of kinetic links that fully describe a kinetic graph built for networking purpose. Moreover, it also guarantees the best link between two nodes, with respect to the target graph algorithm, is always chosen and activated at each time instant.

### 2.4 Aperiodic Neighborhood Maintenance

A limitation in per-event maintenance strategies is the neighborhood maintenance. While mobility prediction and the kinetic graph approach allow to discard invalid links or unreachable neighbors, it remains impossible to passively acquire new neighbors reaching some other nodes’ neighborhood. The lack of an appropriate method to tackle this issue would limit Kinetic Graphs’ ability to obtain up-to-date links and effective kinetic weights. We developed several heuristics to help Kinetic Graphs detect nodes stealthily entering some other nodes transmission range in a non-periodic way.

- **Constant Degree Detection**— Every node tries to keep a constant neighbor degree. Therefore, when a node \(i\) detects that a neighbor actually left its neighborhood, it tries to acquire new neighbors by sending a small advertising message.

- **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 5(a), 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 5(b));

A different approach we suggest is identical to the information exchange period proposed in [9], where the probability that a new neighbor moves into the transmission range of node \(u\) within a time interval of \(t\) is computed.

![Figure 5: Heuristics to detect incoming neighbors in a per-event basis](image)

### 3. CONCLUSION

In this paper, 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 mobile topology routing structure without relying on periodic maintenance. For that matter, we provided guidelines for adapting any localized MANET protocol to the kinetic approach. The kinetic graph approach requiring geo-localization information, we described a common packet format for their exchange during a neighborhood discovery process. Then, a trajectory representation must be defined, based on which kinetic link criteria are generated. Finally, we proposed different solutions to aperiodically maintain the neighborhood. The interesting feature of the proposed framework is that the approach is independent of the network model, the criteria chosen to build the backbone, or the localized MANET protocol used, and various approaches or combinations may be tested.

### 4. REFERENCES


