Construction d'une carte coopérative dans les réseaux véhiculaires

Hermes Pimenta de Moraes Jr, Bertrand Ducourthial

Construction d’une carte coopérative dans les réseaux véhiculaires

Hermes Pimenta Moraes Júnior [†] et Bertrand Ducourthial

Sorbonne universités,
Université de Technologie de Compiègne
CNRS UMR 7253 Heudiasyc
CS 60319 - 60203 Compiègne cedex, France

Dans les réseaux véhiculaires, l’identification des voisins est le point de départ de nombreuses applications. De même, la découverte des voisins à plusieurs sauts est nécessaire pour de nombreuses applications ITS [‡] coopératives. Cependant, en raison de la dynamique de ces réseaux, cette tâche n’est pas simple. Généralement, les informations relatives aux nœuds deviennent rapidement obsolètes. En plus, elles peuvent avoir été transmises par des nœuds non fiables. Dans ce contexte, un nœud doit évaluer la confiance qu’il a dans l’information reçue. Nous proposons un algorithme distribué pour la construction coopérative d’une carte fournissant les coordonnées des nœuds et les services disponibles dans le voisinage jusqu’à n sauts. La carte comprend également une estimation de la confiance dans les informations collectées ainsi que la fiabilité des chemins vers les services découverts. Les expériences par émulation de réseau démontrent l’intérêt de notre approche. Elle devrait permettre de sélectionner des vehicules pertinents dans le cadre d’applications ITS coopératives.

Mots-clefs : Vehicular Ad hoc Networks (VANETs), carte dynamique, découverte de service, confiance distribuée

1 Introduction

Learning about neighbors, in the vicinity and in distant areas, is really useful for ITS applications. It can help on cooperative applications focused on traffic safety (danger prevention at road junctions, round-abouts or crossroads...) and infotainment (weather forecast, chatting...).

This paper deals with the construction of a map of neighbors, up to n hops of distance, with their expected GPS positions and proposed services. In order to build such a map, information needs to be propagated from node to node by means of a distributed and cooperative algorithm. In this context, an important notion is the trust in the received information [CSC11].

Trust is considered here as the degree of subjective belief in nodes information ; it is represented as a variable between 0 and 1. Due to the dynamic behavior of VANETs, information related to nodes usually becomes rapidly obsolete. As more an information is forwarded, as more the delay and the distance from the sources increase, augmenting the obsolescence. Also, when it is forwarded by unknown and possibly dishonest vehicles, trust in the information should decrease. Hence, trust should decrease both in time and distance.

On the contrary, collaboration may help to reinforce the trust into a received information. If several neighbors agree to an information, trust in it should be increased. However, reinforcing the trust according to received messages may lead to the so-called *data-incest* [ČMOC13] : the trust into an information is reinforced due to several receptions while there is a single source at the beginning. To avoid such a behavior in the trust computation, we only consider as multiple sources of an information, nodes at one hop of distance.

[†] Hermes Moraes is on leave from Federal University of Lavras - UFLA - while conducting his PhD research at UTC.

[‡]. ITS : Intelligent Transportation Systems.

Our strategy controls therefore both trust decreasing and increasing depending on the topology. In order to avoid rapid and large variations in the trust metric, we added a smoothing technique at the final stage of trust computation. We summarize our approach with 4 design rules :

— **Rule 1 - Newness increases trust** : As more recent an information is, as more correct it is. Trust should be decreased as the data get older.

— **Rule 2 - Distance decreases trust** : Forwarding messages cause delays and may cause errors. So, trust should be decreased at every hop performed by the information.

— **Rule 3 - Multiple sources increase trust** : When several 1-hop neighbors confirm the same data, the trust in it should be increased.

— **Rule 4 - Trust smooth variation** : Trust should vary smoothly in order to be usable.

2 CNM algorithm

Hereafter we detail the key points of our CNM (Cooperative Neighborhood Map) algorithm. It satisfies the previous rules and provides the information required to maintain a dynamic map of neighbors and services. The map is constructed with nodes' id, GPS coordinates, available services, trust and path reliability.

2.1 The `merge` function

CNM algorithm relies on local *views* introduced in [DKP10]. A *n-view* of a node is the list of its neighbors up to n hops, ordered by distance. This is then a list of sets of neighbors. For instance, by taking Figure 1 in its second scenario (in the middle) as reference, a *3-view* of w_2 is $\{w_2\}, \{w_1, w_3\}, \{v_1\}, \{v_2, v_4\}$. Such views can be computed on the fly as follows :

— Each node keeps its *0-view* composed by itself : $\{w_2\}$ for node w_2.

— On the reception of neighbors' views, they are shifted to the right (because the distance is increased by one more hop).

— Then, they are merged together and with the local *0-view*. For instance, if w_2 receives the *1-views* $(\{w_1\}, \{v_1, w_2, w_3\})$ and $(\{w_3\}, \{v_1, w_1, w_2\})$, from w_1 and w_3 respectively, the merging process would be :

$$\begin{array}{lll}
\{w_2\} & & \\
\{\emptyset\}, & \{w_1\}, & \{v_1, w_2, w_3\} \\
\{\emptyset\}, & \{w_3\}, & \{v_1, w_1, w_2\} \\
\hline
\{w_2\}, & \{w_1, w_3\}, & \{v_1, w_1, w_2, w_3\}
\end{array}$$

— The resulting list is simplified by deleting nodes appearing more than once, keeping the closest one (i.e. the leftmost). The process gives then : $\{w_2\}, \{w_1, w_3\}, \{v_1\}$

The resulting list is a new, and more complete view (here a *2-view*) for the node w_2. This local view will then be shared by means of periodic sending actions. Since only bounded views are considered (limited at n hops), the algorithm converges rapidly despite transient failures. It reacts well in case of changes in the network [DKP10].

2.2 Trust Computation

Every CNM node v maintains an array $T_v[\quad]$ of trusts it puts in nodes it has learnt about. This array is forwarded to other neighbors along with the views.

In order to implement the rules defined in Section 1, a time discount denoted by α is applied when the timer expires and no new message has been received for the referenced node. The distance discount denoted by β is applied at every message reception. Finally, the trust metric is reinforced in case of several sources confirming the data.

Let consider Figure 1 again. When w_1 receives a message from v_1, it computes its trust in v_1 by applying the discount β on the received trust (which is 1 because v_1 trusts itself). It then stores $T_{w_1}[v_1] = \beta \times 1 = \beta$. Similarly, w_2 will store $T_{w_2}[w_1] = \beta$ after receiving a message from its 1-hop neighbor w_1. Nonetheless, if the message sent by w_1 also contains w_1's trust in v_1, w_2 will discount it by β and by its own trust in the sender w_1. Node w_2 then stores

$$T_{w_2}[v_1] = \beta \times (T_{w_2}[w_1] \times T_{w_1}[v_1]) = \beta^3 \tag{1}$$

Now, suppose that several nodes (w_1, w_3, w_4) inform w_2 about v_1 (high density scenario). The trust reinforcement for v_1 is estimated by combining the complement of the trust in each sender ($1 - T_{w_2}[w_1]$ to $1 - T_{w_2}[w_4]$) similarly to probabilities, giving : $1 - (1 - T_{w_2}[w_1]) \times (1 - T_{w_2}[w_3]) \times (1 - T_{w_2}[w_4])$. In such a situation, several messages for a given 2-hops neighbor are received from several 1-hop neighbors. We use a selection function denoted $F()$ that returns the selected data [§]. We then obtain the generalized equation :

$$T_a[c] = \beta \times \left(1 - \prod_{i=1}^{n}(1 - T_a[b_i])\right) \times F(T_1, T_2, \ldots, T_n) \tag{2}$$

where a is the receiver, c is the recognized neighbor and T_i is the trust received from the sender node b_i. This equation combines the distance discount β (first term), the multiple sources reinforcement (second term), and the sender's trust in the 2-hops neighbor (third term). Note that Eq. (2) is similar to Eq. (1) when there is a single neighbor, $F(T_1) = \beta$. Indeed, in this case, F() returns β, the trust of b regarding c ($T_b[c]$). It is important to note that our multiple source reinforcement technique prevent any data incest [ČMOC13].

2.3 Trust smoothing

Equation 2 gives a punctual result for the node's trust. Nonetheless, this value may present large variations, specially in unreliable scenarios. Hence, for the sake of a smoothed metric, the punctual result is inserted in a variable sliding window of trust measures. The final trust value is then the average of values within the window.

Knowing that messages losses are the main responsible for the variation, trust window's size is defined in relation to the loss rate. As more messages are lost (generating larger variations), as wider is the trust window (more values are used to obtain the average).

Considering p the probability of a unique message loss. The aggregate probability of receiving at least one message in m messages sent is given by $1 - p^m$. In order to ensure at least one message in the window with a probability q, it is obtained : $1 - p^m = q$. Hence, the trust window's size m, in number of messages, can be defined according to the estimated loss rate p and a fixed probability of insurance q, by the Equation 3.

$$m = \left\lceil \frac{\ln(1-q)}{\ln p} \right\rceil \tag{3}$$

2.4 Reliability estimation

Similarly to the trust, each node v maintains an array $R_v[\quad]$ of reliabilities it estimates for nodes it has learnt about. To perform this estimation, Node v inserts the sequence numbers of messages from u, whatever was the path they used to reach v, into a fix sliding window.

At every timeout, the window is shifted (dropping the last sequence number) and the reliability is computed by dividing the length of the window (number of messages received) by the number of total messages sent (estimated with basis on the maximum sequence number received). This value gives the communication reliability between v and u, also used for smoothing the trust computation (parameter p).

3 Simulated experiments and results

We present here results obtained in a dynamic scenario where two flows of vehicles merge in a highway junction. Three different network densities were used (Minimum, Intermediate, High) in experiments without packet losses and then with a loss rate p of 40%. Figure 1 shows the connection graphs for each density.

The Airplug framework [BDEAK10] was used to implement the CNM algorithm and to carry on the experiments. The main goal was to show the ability of our algorithm to construct a map of neighbors while evaluating the trust of each identified node. Figure 2 illustrates CNM's resulting maps (First, Minimum Density without and with losses, then High Density without and with losses) obtained when using $\beta = 0.8$. Due to the lack of place, we present results only for vehicle w_2 (displayed in blue).

[§]. In order to focus on the trust computation, we simply selected the most recent information based on the sequence numbers in the presented results, though a more complex strategy could be used here.

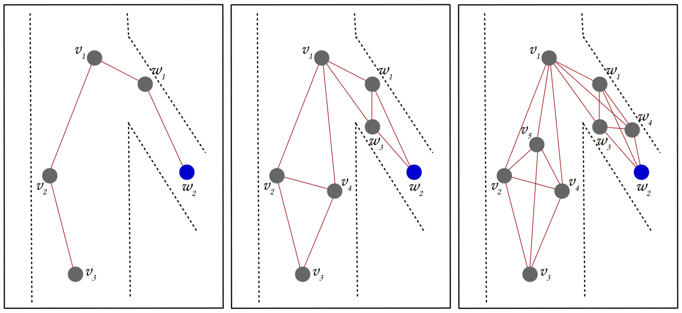
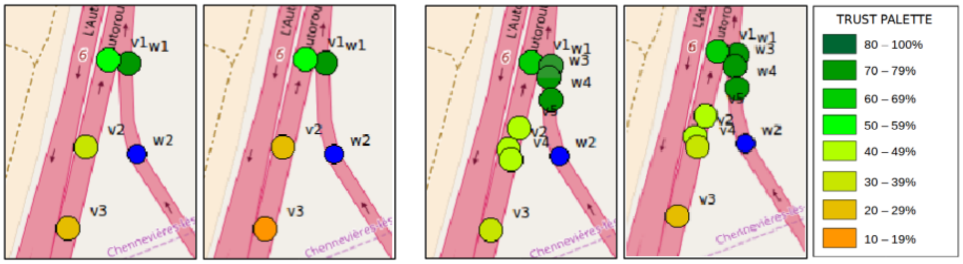


FIGURE 1: Connectivity graphs for the highway scenario. Minimum, Intermediate and High densities.



(a) Without and with losses scenarios

(b) Without and with losses scenarios

FIGURE 2: Resulting maps for Minimum density (left), and High density (right)

It can be seen in the resulting maps that w_2's trust on nodes decreases as the distance (in hops) increases (Rule 2 acting). This behavior is more evident in scenarios with losses. In this case, the trust is decreased when no new message is received (Rule 1 acting). Finally, we can say that losses are less relevant in high density scenarios (there are more nodes to forward a message) where higher values were achieved for the trust metric.

4 Conclusions

CNM was proposed with the main goal of constructing a general dynamic map of neighbors up to n hops. A distributed and cooperative strategy where neighbors' trust decreases in time and distance from the data source and increases in case of multiple sources reinforcement was developed. The strategy avoids data incest with an approach based on graphs and offers an evaluation of the multi-hop path quality towards each identified node. It is expected that the resulting map can be used by latter applications in order to choose nodes to communicate with, in cooperative ITS applications.

Références

[BDEAK10] Anthony Buisset, Bertrand Ducourthial, Farah El Ali, and Sofiane Khalfallah. Vehicular networks emulation. In *ICCCN 2010*, Switzerland, Aug 2010.

[ČMOC13] J. Čurn, D. Marinescu, N. O'Hara, and V. Cahill. Data incest in cooperative localisation with the common past-invariant ensemble kalman filter. In *Proceedings of the 16th International Conference on Information Fusion*, pages 68–76, July 2013.

[CSC11] J. H. Cho, A. Swami, and I. R. Chen. A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys Tutorials*, 13(4) :562–583, Fourth 2011.

[DKP10] Bertrand Ducourthial, Sofiane Khalfallah, and Franck Petit. Best-effort group service in dynamic networks. In *Proceedings of the twenty-second annual ACM symposium on Parallelism in algorithms and architectures*, pages 233–242. ACM, 2010.