The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

POMDP-Based Decision Making for Fast Event Handling in VANETs

Shuo Chen, Athirai A. Irissappane,† Jie Zhang
School of Computer Science and Engineering, Nanyang Technological University, Singapore
†University of Washington, Washington, USA
chen1087@e.ntu.edu.sg, athirai@u.washington.edu, zhangj@ntu.edu.sg

Abstract

Malicious vehicle agents broadcast fake information about traffic events and thereby undermine the benefits of vehicle-to-vehicle communication in vehicular ad-hoc networks (VANETs). Trust management schemes addressing this issue do not focus on effective/fast decision making in reacting to traffic events. We propose a Partially Observable Markov Decision Process (POMDP) based approach to balance the trade-off between information gathering and exploiting actions resulting in faster responses. Our model copes with malicious behavior by maintaining it as part of a small state space, thus is scalable for large VANETs. We also propose an algorithm to learn model parameters in a dynamic behavior setting. Experimental results demonstrate that our model can effectively balance the decision quality and response time while still being robust to sophisticated malicious attacks.

Introduction

Nowadays people rely on map services such as Google Maps for better driving experience. Map services are mostly centralized resulting in information update delay. Events affecting traffic condition (e.g., car accidents, road construction) require fast information update to avoid traffic congestion. Vehicular Ad-hoc Networks (VANETs) allow vehicle (agents) to communicate directly with each other and support fast information exchange (Hartenstein and Laberteaux 2008). When a vehicle directly observes an event, it can broadcast event alerts through vehicle-to-vehicle communication, based on which other vehicles can decide to respond.

Nevertheless, selfish/malicious vehicles (Lin, Kraus, and Shavitt 2007; Raya and Hubaux 2007) exist which share fake/untrustworthy information to mislead other drivers. Trust management schemes (Raya et al. 2008; Chang and Kuo 2009; Li et al. 2012; Wei et al. 2014; Li and Song 2016) address this issue by computing the trustworthiness of vehicles and thereby the trustworthiness of messages exchanged about an event. Such information helps to create and update a vehicle's belief, i.e., probability distribution about whether the traffic event occurred or not (more messages about the event result in a higher belief). However, existing trust schemes only focus on trust computation but do not effectively reason about driving decisions[1], e.g., whether to re-route or stay on the same route. These schemes use a simple heuristic way and decide to re-route, if the belief about the occurrence of an event (called event belief, hereon) exceeds a given threshold. Often, thresholds are set to a larger value for better decision-making. This requires exchanging more messages, resulting in higher resource consumption (communication bandwidth, computation cost), as well as longer information gathering period leading to delayed response, which can add to the severity of the traffic event.

To achieve fast response and reduce resource consumption without sacrificing decision quality, we propose a Partially Observable Markov Decision Process (POMDP) based approach. POMDP provides a natural framework for sequential decision making under uncertainty. Thus, it is an ideal choice for VANETs to handle the uncertainty introduced due to unreliable information from malicious vehicles. Besides, POMDP effectively balances the trade-off between exploitation (deciding to re-route or not) and exploration (collecting more information about the event) (Kaelbling, Littman, and Cassandra 1998).

Our POMDP model maintains belief about the occurrence of a traffic event and assists a vehicle agent to decide whether to re-route or not based on its current belief. It gathers information about the event by querying neighboring vehicles (which can be trustworthy/untrustworthy in providing information) and updates the event belief accordingly. Based on the belief, the POMDP model balances the expected benefit of re-routing and the delay/cost of querying more information, thereby maximizing the expected total utility for the vehicle. The environmental parameters (observation, transition functions) for the POMDP model need to be learned. In our scenario, learning is more complex as the observation function representing the behavior of neighboring vehicles changes over time. This is because of sophisticated malicious attackers who can change their behavior periodically. We have addressed the above challenges and made the following contributions in this paper: 1) To the best of our knowledge, this is the first time POMDP is applied for the problem; 2) While the POMDP model can effectively balance the trade-off between decision quality and delay/cost,

[1]While there are many other driving decisions, we will use re-routing to explain our approach in this paper.

it can also handle the issue of fake information. Further, the model is designed to have a small state space and thereby scalable to a large number of vehicles; 3) We have also proposed an algorithm based on Bayesian reinforcement learning for learning the observation function in a dynamic scenario where malicious vehicles change their behavior frequently; 4) We have conducted extensive experiments to evaluate the performance of our model. Results demonstrate that our solution can make better and fast decisions while still being robust under frequent behavior change.

Related Work

Several works have been proposed to model the trustworthiness of vehicles. Chang and Kuo (2009) design a Markov Chain Trust Model in which the steady trust-state transition probability is regarded as vehicle's trustworthiness. Wei et al. (2014) and Li and Song (2016) use the Dempster-Shafer Theory to combine own experience and the information from neighbors. Any such trust scheme can be used to derive vehicle's trustworthiness for our POMDP model.

Some existing works utilize the trustworthiness of vehicles and their messages to determine the occurrence of a traffic event. Raya et al. (2008) propose a data-centric trust scheme to compute the trustworthiness of each message received based on sender's trustworthiness, time and location. The event belief is obtained by aggregating the messages using voting, Bayesian inference, or Dempster-Shafer Theory. More messages reduce the uncertainty in event belief while delaying the response. Li et al. (2012) propose a reputation-based announcement scheme in which the exchanged message contains event information and sender's trustworthiness. If the trustworthiness is greater than a preset threshold, vehicles will believe the event immediately. A smaller threshold value enables faster response but increases the probability of being misled. Our POMDP model, on the other hand, avoids the complexity of fine tuning thresholds.

We also notice that in other domains, there are existing works integrating POMDP with trust management for decision-making (Irissappane, Oliehoek, and Zhang 2014; Irissappane et al. 2015; Irissappane, Oliehoek, and Zhang 2016). However, they model the trustworthiness of all entities in the system as POMDP states, resulting in a large state space and thereby not scalable for VANETs. Although the work in (Irissappane, Oliehoek, and Zhang 2016) tries to solve the scalability issue by splitting a large state space into smaller identical subsets, this technique becomes more complex in the case of VANETs as the subsets might not be identical. In our model, we only consider the trustworthiness of the sender who sends the current message, thereby limiting the size of the state space. Further, in the above approaches the POMDP parameters are manually set, while we learn the POMDP parameters in our model.

Background

Here we introduce the background on POMDP and describe a general method to learn its unknown model parameters.

POMDP

Partially Observable Markov Decision Process (POMDP) can be formulated as a tuple $\langle S, A, T, R, \Omega, O\rangle$, in which S is a set of states[2]; A is a set of actions which cause the transition of current state s to next state s'; T is the state transition function specifying the probability distribution over the next states $Pr(s'|s,a)\ \forall\ s' \in S$, given the current action a and current state s; R is the reward function which returns the expected reward $R(s,a)$; Ω is a set of observations obtained after taking an action; O is the observation function specifying the probability distribution over observations $Pr(o|s',a)\ \forall\ o \in \Omega$, given a and s'. POMDP maintains a belief state b which is a probability distribution over states. If $b(s)$ represents the probability that the world is in state s, the updated belief state $b'(s')$ is calculated every time any action a is taken and observation o is received,

$$b'(s') = Pr(s'|o,a,b) = \frac{Pr(o|s',a)}{Pr(o|b,a)} \sum\nolimits_s Pr(s'|s,a)b(s)$$

As $Pr(o|b,a)$ is a normalization factor, the belief update is:

$$b'(s') \propto Pr(o|s',a) \sum\nolimits_s Pr(s'|s,a)b(s) \tag{1}$$

A POMDP policy is the mapping from belief state to action. The objective of a POMDP model is to find the optimal policy which can maximize the expected reward. The expected total reward obtained by executing a policy from a belief state is specified by a value function, details of which can be found in (Kaelbling, Littman, and Cassandra 1998).

Bayesian Reinforcement Learning for POMDP

While POMDP provides a principled framework for planning under uncertainty, the model parameters are often unknown and need to be learned during interaction with environment. Bayesian reinforcement learning techniques (Ghavamzadeh et al. 2015) have been studied in literature to learn these parameters in which Dirichlet distribution is used to maintain a probability distribution over the possible model parameters.

If $O^{s'a}$ represents the probability distribution over observations Ω given state s' and action a, we assume the prior to be a Dirichlet distribution $Dir^{s'a}(\phi^{s'a}_{o_i}|i = 1,\ldots,|\Omega|)$, where $\phi^{s'a}_{o_i}$ is the Dirichlet parameter. The probability that observation o_i is obtained given state s' and action a can be computed as (Ross, Chaib-draa, and Pineau 2008):

$$Pr(o_i|s',a) = \frac{\phi^{s'a}_{o_i}}{\sum_{j=1}^{|\Omega|} \phi^{s'a}_{o_j}} \tag{2}$$

If $\vec{h}^{s'a}$ denotes the observation history, i.e., a set of observations when action a causes a transition to state s', then the posterior of $O^{s'a}$ is expressed as a Dirichlet distribution $Dir^{s'a}(\phi^{s'a}_{o_i} + N(o_i|\vec{h}^{s'a})|i = 1,\ldots,|\Omega|)$, where $N(o_i|\vec{h}^{s'a})$ is the number of times observation o_i appears in $\vec{h}^{s'a}$. The probability $Pr(o_i|s',a)$ is then given by,

$$Pr(o_i|s',a) = \frac{\phi^{s'a}_{o_i} + N(o_i|\vec{h}^{s'a})}{\sum_{j=1}^{|\Omega|}[\phi^{s'a}_{o_j} + N(o_j|\vec{h}^{s'a})]} \tag{3}$$

[2]We only consider discrete sets of states and observations.

In general, we do not exactly know the state s' after receiving an observation in POMDPs. However, it is to be noted that in VANETs, state s' can be determined (after a certain time period) as ground truth can be obtained from external sources such as a centralized map system.

System Model

Server is in charge of calculating trustworthiness and issuing certificates for vehicles. The certificate supports identity verification and information integrity checking (Kargl et al. 2008). It also contains the trustworthiness score[3] of the vehicle. The server periodically generates a new certificate for each vehicle which expires after a given time period. Vehicles can download new certificates when driving by the Road Side Units (RSUs) (Sun et al. 2010). When exchanging messages, vehicles attach certificates to prove their identity and trustworthiness. Note that certificates cannot prevent vehicles from sharing false information.

The server maintains and updates vehicles' trustworthiness. Vehicles can upload received messages to the server through RSUs. We assume the server is integrated with a centralized map service which provides the ground truth about an event, though with some delay. It then compares each uploaded message with the ground truth to determine the trustworthiness of the vehicle sending that message.

Vehicles exchange local event information with their neighbors. Apart from broadcasting *event alerts*, vehicles (called *requestors*) are allowed to query about uncertain events[4] from neighbors to gather more information and thereby make more informed decisions. Vehicles which broadcast event alerts or reply to queries from neighbors are termed *reporters*. Each vehicle is also equipped with centralized map service, whose information update can be delayed. However, it can still provide the true world state, which will be useful to learn the observation function as explained later.

Information sharing is achieved by broadcasting messages. The messages can be received by all the vehicles inside the communication range of senders. Messages can be of three types, event alerts, queries or responses to queries. Each message includes the event location and observed time. *Requestors* query an event by including the event location and time in the query message. The response messages further contain *reporters*' opinion about the event. Vehicles close to the event location can directly observe the event and respond to queries about the event. However, malicious vehicles may send untruthful opinions and even broadcast fake alerts about nonexistent events. Once a vehicle receives an alert or response message, it adds the message to a unique message queue if the event is in future route and will affect its driving experience. Otherwise, it discards the message.

Proposed POMDP Model

We now present the proposed POMDP model and the algorithm for learning the observation function in the model.

[3]Though trustworthiness is a continuous value, it is discretized to fit our POMDP model and use standard POMDP solvers.

[4]The events have already occurred but are not directly visible.

Model Design

Our POMDP model aids vehicle agents in fast and efficient decision-making during unforeseen traffic events. It makes quality decisions in a timely manner using less number of messages (queries, responses) with neighboring vehicles. The model maintains a belief, i.e., probability distribution about whether a traffic event has occurred. It updates this belief by processing the messages about the event. The model also assumes that the neighboring vehicles can exhibit different behaviors, i.e., truthful, malicious etc., in providing event information. Based on the event belief, the model then decides to re-route or stay on the current route.

We consider only a single event per POMDP execution as it is impossible to decide the number of unforeseen events in advance. For handling multiple events, we can simply initialize a POMDP for each of them. The POMDP components are described in detail below.

State The POMDP state $s = \langle r, l, f \rangle$ is composed of variables representing the occurrence of a traffic event $r \in \{yes, no\}$, the trustworthiness $l \in \mathcal{L} : \{high, medium, low\}$ of the neighboring vehicle (called sender) which sent the current message and the freshness $f \in \mathcal{F} : \{fresh, old\}$ of the sender's trustworthiness value. Malicious vehicles might use old certificates as their behavior often results in lower trustworthiness value in their new certificates, which is why freshness is included as a part of the state space. We only model the trustworthiness of the sender who sent the current message and perform belief updates based on each message. This avoids the complexity that arises due to large state space which often results in scalability issues for POMDP.

Action Our model knows the following actions: 1) $query$ (Q) for collecting event information from neighbors. Query action represents checking the unique message queue for the traffic event. If messages are present, they are retrieved in First-in-First-out order and serve as observations for our POMDP model. If the queue is empty, a query message is broadcast to neighbors; 2) $reroute$ (RE) for changing the current route in order to avoid the traffic event; 3) $ignore$ (IG) for ignoring the event and staying on the current route.

State Transition Function The function is formulated as $Pr(s' = \langle r', l', f' \rangle | s = \langle r, l, f \rangle, a)$. For $query$ action, $r = r'$ while l' and f' are unknown before receiving any observation (by retrieving a message from the message queue). Since the message queue could be empty and the observation may not be available immediately, we consider l' and f' to be uniformly distributed in their space. Thus, the transition function for $query$ action is given by:

$$Pr(s'|s, Q) = \frac{\delta_{r,r'}}{|\mathcal{L}||\mathcal{F}|} \tag{4}$$

$\delta_{r,r'}$ is the Kronecker delta which is 1 if $r' = r$, and 0 otherwise. The denominator $|\mathcal{L}||\mathcal{F}|$ represents the uniform distribution over the trustworthiness and freshness levels.

When $reroute$ or $ignore$ action is performed, the decision-making process for this event is complete and we can simply set their transition function as a uniform distribution, $Pr(s'|s, RE/IG) = \frac{1}{2|\mathcal{L}||\mathcal{F}|}$.

Reward Function Since query message results in resource consumption and delays response to event, it incurs cost $R(r, Q)$. On making a correct decision (i.e., $reroute$, when traffic event exists and $ignore$ otherwise), rewards $R(r = yes, RE)$ and $R(r = no, IG)$ are given, respectively. For a wrong decision (i.e., $ignore$, when traffic event exists and $reroute$ otherwise), there are penalties $R(r = no, RE)$ and $R(r = yes, IG)$.We assume that rewards have a positive value and penalties have a negative value and there is more impact when a traffic event occurs ($r = yes$), thus,

$$R(r = yes, RE) = -R(r = yes, IG)$$
$$R(r = no, IG) = -R(r = no, RE)$$
$$|R(r, Q)| < R(r = no, IG) < R(r = yes, RE)$$

Observation The received alerts and response messages are the observations in our model. As each message contains the sender's opinion about the traffic event, the trustworthiness of sender and the time stamp indicating when the trustworthiness is updated (which determines the freshness level of trustworthiness), the observation is denoted as a tuple $o = \langle o_r, o_l, o_f \rangle$ where $o_r \in \{yes, no\}$ is the sender's opinion about the traffic event, $o_l \in \mathcal{L}$ is the sender's trustworthiness and $o_f \in \mathcal{F}$ is the freshness of trustworthiness.

Observation Function In our model, we assume full observability on the trustworthiness of sender and its freshness as they are values contained in the certificate. Only the occurrence of the traffic event, i.e., r is partially observable. We formulate our observation function to reflect this. For the $query$ action Q,

$$Pr(\langle o_r, o_l, o_f \rangle | \langle r', l', f' \rangle, Q) \quad (5)$$
$$= \delta_{o_l, l'} \cdot \delta_{o_f, f'} \cdot Pr(o_r | \langle r', o_l, o_f \rangle)$$

where the Kronecker delta $\delta_{o_l,l'} = 1$, if $o_l = l'$ and $\delta_{o_f,f'} = 1$ if $o_f = f'$. Their value is 0, otherwise. $Pr(o_r|\langle r', o_l, o_f \rangle)$ represents the probability that the sender responds with the opinion o_r given the ground truth about the event r' and (o_l, o_f). It is to be noted that neighboring vehicles with the same trustworthiness and freshness levels can have different $Pr(o_r|\langle r', o_l, o_f \rangle)$ values. Our model considers the observation function to represent the behavior of all neighboring vehicles rather than one single vehicle. Further, $Pr(o_r|\langle r', o_l, o_f \rangle)$ in the observation function is unknown and needs to be learned. For $reroute$ and $ignore$ action, we set their observation function as uniform distribution as they do not contribute in gathering event information.

Belief Update

Given the observation $o = \langle o_r, o_l, o_f \rangle$, current state $s = \langle r, l, f \rangle$ and the next state $s' = \langle r', l', f' \rangle$, the belief update based on Eqn. 1 is given by,

$$b'(\langle r', l', f' \rangle) \propto Pr(o|s', Q) \sum\nolimits_s Pr(s'|s, Q) b(s)$$

From Eqn. 5, $Pr(o|s', Q) = 0$, when $l' \neq o_l$ or $f' \neq o_f$. Thus $b'(\langle r', l', f' \rangle) = 0$ for the same cases. Let $b'(r')$ denote the next belief on r', we have:

$$b'(r') = \sum_{l' \in \mathcal{L}, f' \in \mathcal{F}} b'(\langle r', l', f' \rangle) = b'(\langle r', o_l, o_f \rangle)$$
$$b'(r') \propto Pr(o_r | \langle r', o_l, o_f \rangle) \sum\nolimits_s Pr(s'|s, Q) b(s)$$

Substituting Eqn. 4 and note the $\delta_{r,r'}$, we further have,

$$\sum_s Pr(s'|s, Q) b(s) = \sum_{l \in \mathcal{L}, f \in \mathcal{F}} \frac{1}{|\mathcal{L}||\mathcal{F}|} b(\langle r', l, f \rangle) \quad (6)$$

$b(\langle r', l, f \rangle) \neq 0$, when $l = l_p$ and $f = f_p$, where l_p and f_p denote the trustworthiness and freshness of the previous message sender. Thus,

$$\sum_s Pr(s'|s, Q) b(s) = \frac{1}{|\mathcal{L}||\mathcal{F}|} b(\langle r', l_p, f_p \rangle) \quad (7)$$

If $b(r')$ is the current belief on r',

$$b(r') = \sum_{l \in \mathcal{L}, f \in \mathcal{F}} b(\langle r', l, f \rangle) = b(\langle r', l_p, f_p \rangle)$$

Substituting for $b(\langle r', l_p, f_p \rangle)$ in Eqn. 7,

$$\sum_s Pr(s'|s, Q) b(s) = \frac{1}{|\mathcal{L}||\mathcal{F}|} b(r')$$

Thus the belief update is given by:

$$b'(r') \propto Pr(o_r | \langle r', o_l, o_f \rangle) \cdot b(r') \quad (8)$$

Note that Eqn. 8 is actually Bayesian inference. Therefore, our POMDP model implicitly incorporates Bayesian inference along with the decision-making process.

Learning the Observation Function

Existing methods for learning the POMDP parameters try to improve learning speed and lower approximation error (Ghavamzadeh et al. 2015). However, most of them assume the POMDP model to be stationary and thus are not suitable for VANETs, as neighboring vehicles change with time and hence their behavior model (represented by the observation function) also changes. Existing works need longer learning periods to override historical data before adapting to the changed model. To address this issue, more importance can be given to recent experience (Jaulmes, Pineau, and Precup 2005), however, doing so will ignore valuable history information which could be useful in settings where malicious vehicles change their behavior frequently.

Here, we propose Algorithm 1 to learn the observation function in our POMDP model under dynamic conditions where the behavior of neighboring vehicles changes. We know from Eqn. 5 that learning the observation function implies learning the probability distribution $Pr(o_r|\langle r', o_l, o_f \rangle)$. For each $s' = \langle r', o_l, o_f \rangle$, we use a Dirichlet distribution $Dir^{s'}(\phi^{s'}_{o_r}, o_r \in \{yes, no\})$ to maintain the prior over $Pr(o_r|s')$. Note that the Dirichlet distribution here is actually a Beta distribution since o_r only has

1 Initialize model P_{cur}, history set H and time window W.
2 **while** *driving* **do**
3 **if** *new* experience e *is received* **then**
4 Initialize a new Dirichlet distribution $Dir^{s'}(\phi_{o_r}^{s'})$ and compute its posterior using e;
5 Compute $\vec{v_e}$ from $Dir^{s'}(\phi_{o_r}^{s'})$ (Eqn. 9) and add it to H;
6 Cluster the vectors in H using K-Means;
7 Find the cluster C which contains $\vec{v_e}$;
8 Find its cluster center $\bar{C}$;
9 Initialize a POMDP model P with *observation function* as $\bar{C}$;
10 Set P_{cur} to be P;

Algorithm 1: Learning the Observation Function

two possible values. Given the observation history $\vec{h}^{s'}$, the posterior $Pr(o_r|s')$ can be computed by,

$$Pr(o_r|s') = \frac{\phi_{o_r}^{s'} + N(o_r|\vec{h}^{s'})}{\sum_{j=1}^{|\Omega|}[\phi_{o_j}^{s'} + N(o_j|\vec{h}^{s'})]} \quad (9)$$

To handle behavior change, we periodically collect observation histories. Vehicles know the ground truth about the occurrence of traffic event from the centralized map service after some delay. The observations obtained in the form of responses and alerts from neighboring vehicles as well as the ground truth from the centralized map service are gathered over a stipulated time period and constitute event *experience*. Each *experience* forms a set of observation histories $\{\vec{h}^{s'}|\forall s' \in S\}$ and is used to formulate the posterior $\{Pr(o_r|s')|o_r \in \{yes, no\}, \forall s' \in S\}$ for that time period. With time, vehicles gather a set of *experience*, using which the $Pr(o_r|s')$ for every time period is determined. As $Pr(o_r = no|s') = 1 - Pr(o_r = yes|s')$, the set $\{Pr(o_r|s')|o_r \in \{yes, no\}, \forall s' \in S\}$ can be sufficiently represented using a vector $\vec{v_e} = \{Pr(o_r = yes|s')|\forall s' \in S\}$, and the vector length $|\vec{v_e}|$ equals to the size of state space which is $2|\mathcal{L}||\mathcal{F}|$. Thus, we will obtain $\vec{v_e}$ for each experience e gathered in every time period. We then use K-Means clustering to cluster $\vec{v_e}$ across different time periods based on Euclidean distance. Doing so allows us to determine the recurrent behavior patterns of neighboring vehicles. Once the clusters are formed, we locate the cluster C which contains $\vec{v_e}$ representing the most recent *experience/time period*. This cluster represents the current behavior pattern of neighbors. Though this cluster might not be the representative of neighbors who change their behavior immediately after the most recent *experience* was collected, this scenario is rare.

If $\mathcal{V}$ denotes vectors $\vec{v_e}$ in C, and $\{\widehat{Pr}(yes|s')|\forall s' \in S\}$ is the vector derived based on the most recent *experience*, the cluster center $\bar{C}$ can be computed as:

$$\bar{C} = \left\{ \frac{\sum_{i=1}^{|\mathcal{V}|} Pr_i(yes|s') \cdot |\vec{h}_i^{s'}| \cdot \Delta P_i}{\sum_{i=1}^{|\mathcal{V}|} |\vec{h}_i^{s'}| \cdot \Delta P_i}, \forall s' \in S \right\} \quad (10)$$

where $\Delta P_i = 1 - \sqrt{|Pr_i(yes|s') - \widehat{Pr}(yes|s')|}$. $\bar{C}$ represents the current observation function. It can be used to initialize a new POMDP model which will be able to handle the current set of neighbors. $|\vec{h}_i^{s'}|$ in Eqn. 10 normalizes the effect of observation histories of different length. ΔP_i degrades the effect of vectors which are not that similar to $\{\widehat{Pr}(yes|s')|\forall s' \in S\}$. This is meaningful when the preset cluster number is small and thereby the vectors clustered into the same cluster might not be very similar.

The details of Algorithm 1 are described as follows. Let P_{cur} be the POMDP model whose optimal policy is executed to obtain a new *experience* for the current time period, H be the set of vectors $\{\vec{v_e}\}$ collected across all time periods and W denote the time window, i.e., length of each time period. When new experience is obtained, the corresponding $\vec{v_e}$ is derived (lines 4-5), after which the K-Means algorithm is used to determine the clusters (line 6). Each identified cluster corresponds to a behavior pattern. We locate the cluster which contains the recent $\vec{v_e}$, initialize a new POMDP model with the new observation function computed using Eqn. 10, and set P_{cur} as the new model (line 7-10) which advises all the actions to be taken in the next time window. Note that this algorithm requires initializing a POMDP model and computing its optimal policy periodically. This is practical only when the model state space is small, and our design satisfies this condition.

Experiments

We conduct extensive experiments to compare the performance of our POMDP approach with threshold-based trust schemes in terms of decision quality and cost/delay under static behavior settings. Also, we analyze the robustness of our proposed algorithm for learning observation function under dynamic behavior settings.

Simulation Setup

We evaluate our POMDP model using the *Veins* simulator (Sommer, German, and Dressler 2011), which combines OMNeT++ for vehicle-to-vehicle communication simulation, and SUMO for road traffic microsimulation. We periodically set the speed of some vehicles to zero in order to mimic traffic accidents. When a vehicle is close enough (25 meters) to an event location, we deem that the vehicle has directly observed the event and can broadcast alerts or responses. We also add trust management function to the Scenario Manager of Veins simulator to make it behave as server. The initial trustworthiness value of vehicles is 0.5. In POMDP, we regard the trustworthiness as *high* when its value is larger than 0.8, *low* when the value is less than 0.3 and *medium* otherwise. The freshness is *fresh* if the trustworthiness is updated within 30 seconds (s) and *old* otherwise. The values for rewards used in our experiments are: $R(r = yes, RE) = -R(r = yes, IG) = 30, R(r = no, IG) = -R(r = no, RE) = 20$ and $R(r, Q) = 1$. We set $R(r, Q) = 1$ to encourage query when the event belief is highly uncertain. 210 vehicles join the simulation within 120s and ten of them are used to mimic accidents. When a vehicle arrives its destination, it will choose a new random destination. Therefore, each vehicle is active until the simulation ends. Averagely, there is one accident every 35s and

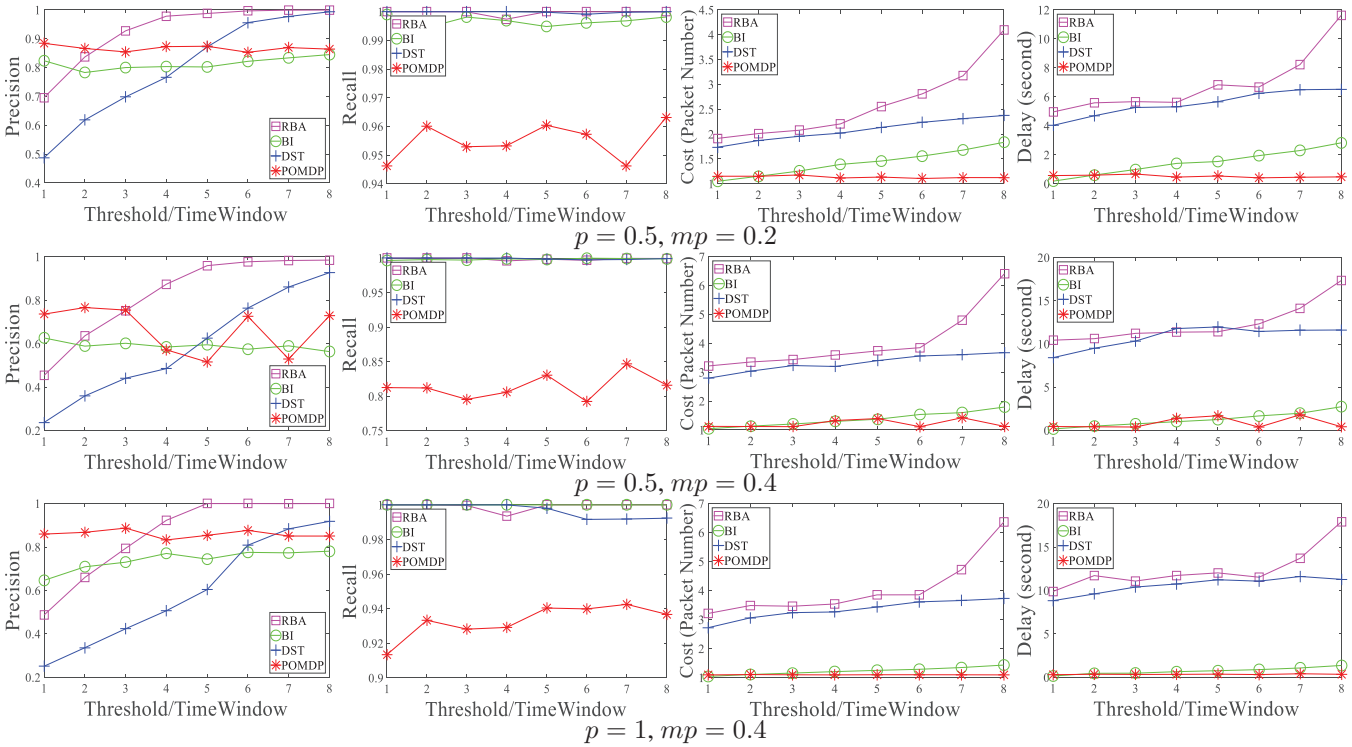


Figure 1: Performance Evaluation under Static Behavior

each accident lasts about 120∼180s. The ground truth update delay from centralized map service is set to 60s.

Our POMDP model is compared with Dempster-Shafer Theory (DST) and Bayesian inference (BI) (Raya et al. 2008) and reputation-based announcement (RBA) (Li et al. 2012). (Raya et al. 2008) maintains two beliefs for each event, i.e., b_r: *event happens* and b_f: *event does not happen*. For (Raya et al. 2008), we compare the difference between b_r and b_f, i.e., $|b_r - b_f|$, with the threshold to decide whether to re-route or not. In RBA, we use the sender's trustworthiness as well as a time-decay factor to make the decision. We use SARSOP (Kurniawati, Hsu, and Lee 2008) to compute the POMDP policy. K-means clustering is performed by Cluster 3.0 software[5]. We evaluate the decision-making performance under both static and dynamic scenarios.

The evaluation metrics are: 1) precision: $\frac{TP}{TP+FP}$ where TP is true positive, means how many times vehicles re-route when events exist and FP is false positive, means how many times vehicles re-route when events are fake; 2) recall: $\frac{TP}{TP+FN}$ where FN is false negative, means how many times vehicles ignore when events exist; 3) decision cost: how many messages are processed before making each decision; 4) decision delay: how long a vehicle needs to make decision after receiving the first message about a certain event.

Malicious Behaviors

In our simulation, we have considered both static and dynamic malicious behaviors described as follows.

[5]http://bonsai.hgc.jp/∼mdehoon/software/cluster/software.htm

Static: Vehicles report truthful opinions with probability p and lie with probability $(1-p)$. The probability p does not change with time.

DynamicI: Malicious vehicles report truthful opinions until their trustworthiness value reaches *high*, after which they provide complementary opinions.

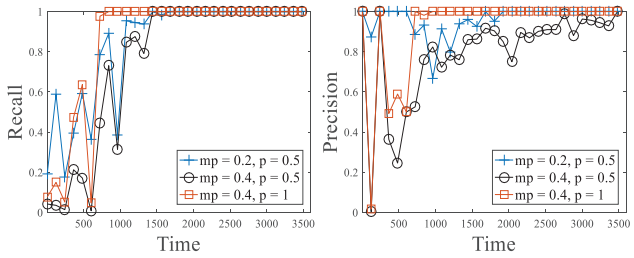
DynamicII: Similar to *DynamicI*, vehicles can report truthful opinions only when their trustworthiness is *low*.

The update of trustworthiness is set as: vehicles will update their trustworthiness in a long interval (60s) if they have lied recently while they will update trustworthiness frequently (every 20s) if they have reported truth.

Static Behavior Scenario

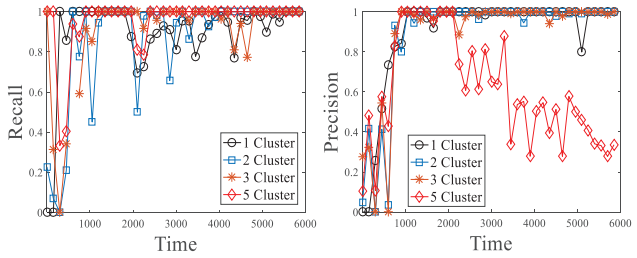
In the static scenario, we have evaluated the performance when malicious vehicles exhibit *Static* behavior with $p = 0.5$ and $p = 1$, and the percentage of malicious vehicles mp is 0.2 and 0.4. The overall driving period is 3600s. The results are as shown in Fig. 1. The performance of threshold-based schemes is evaluated under different thresholds. However, since there is no threshold setting in POMDP, we evaluate the results under different time windows. Thus, the x-axis is threshold being from 0.1 to 0.8 for threshold-based schemes and time window W being from 10s to 80s for POMDP. We set the cluster number of K-Means clustering as 1 for this scenario, i.e. POMDP learns the observation function without detecting behavior patterns.

The Fig. 1 shows that in all the three cases, RBA, BI and DST have low precision and high recall when setting a low threshold. This is because a low threshold value allows high



(a) (b)

Figure 2: The Warm Up Period in POMDP ($W = 60s$)



(a) (b)

Figure 3: Decision Quality under Dynamic Behavior

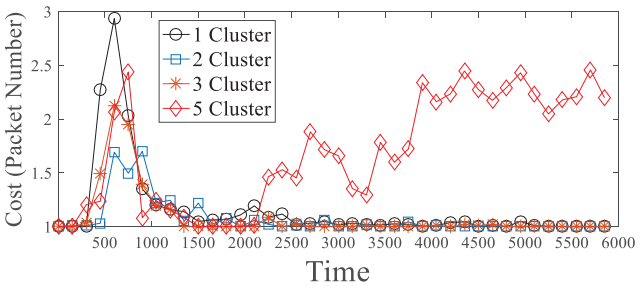


Figure 4: Cost under Dynamic Behavior

uncertainty about event occurrence and vehicles choose to re-route most of the time, i.e. large FP and low FN. When the threshold increases, these schemes seek more messages to reduce uncertainty, and thus have high precision. However, high threshold also incurs more cost/delay for DST and RBA especially when mp is large. Compared to threshold-based schemes, POMDP has the lowest cost/delay under different time windows and different behavior settings. Our POMDP model is more efficient because it uses Bayesian theory to update belief. The opinion from the vehicles with low trustworthiness will strengthen the belief about the opposite of that opinion, i.e. false information also contributes to event belief. Thus, POMDP often just needs one or two messages to achieve high event belief. BI has the second lowest cost/delay due to the same reason.

For decision quality, the POMDP solution has the lowest recall and lower precision when the threshold of DST and RBA is high. This is due to the high FN and FP which occur at the beginning of simulation (called *warm up period*). In specific, at the beginning, the trustworthiness of benign and malicious vehicles are both (*medium*, *fresh*) while malicious vehicles send lots of fake alerts. Thus, the true information from benign vehicles seems untrustworthy and makes vehicles believe in the opposite since POMDP updates belief with Bayesian theory. This effect will disappear after benign and malicious vehicles have different trustworthiness and POMDP learns the $Pr(o_r|s')$ given different s'. We collect precision and recall data in every 120s and show this process in Fig. 2. It shows that recall and precision increase to a high value with time. While different mp and p settings affect this increasing speed, the trend is the same, i.e. POMDP can handle malicious behavior very well through learning. Thus, the decision quality of POMDP is actually at least as good as threshold-based schemes if leaving out the warm up period. Overall, our model has the same if not better decision quality than threshold-based schemes while its decision cost/delay is lowest.

Dynamic Behavior Scenario

In this section, we evaluate the performance of our POMDP model under dynamic behavior setting. In specific, we set the overall simulation period as 6000s, and the precision, recall and cost data are collected every 150s. The percentage of malicious vehicles mp is set as 0.3. Malicious vehicles exhibit *Static* behavior with $p = 1$ till 2000s, after which they exhibit *DynamicII* behavior until 4000s. Then they change the behavior to *DynamicI* after that. We have tested the performance when cluster number equals to 1, 2, 3, and 5. The time window W is set to 60s. The results about precision and recall are as shown in Fig. 3. From Fig. 3a, we can observe that when simulation time is around 2000s, the recall value decreases significantly. This is because the malicious vehicles with *low* trustworthiness start to tell truth, which is totally opposite to their previous behavior. Therefore, when malicious vehicles report a true event, other vehicles will ignore it and FN increases. Comparing the different cluster numbers, it can be observed that recall recovers fast when adopting clustering while POMDP needs more time to adapt to new behavior without clustering. When simulation time reaches 4000s, malicious vehicles change behavior to *DynamicI*. However, the influence is limited because most of the malicious vehicles do not have *high* trustworthiness at that time and they will keep telling truth, which is similar to their previous behavior. From Fig. 3b, we can find that after first behavior change, the POMDP with cluster number being 5 starts to have low precision. This is because the cluster number is too large for the existing *experience*, i.e. each cluster only has several vectors. In this case, the learning algorithm can not catch the current behavior appropriately. This implies that the setting of cluster number should fit with the amount of *experience*.

The cost of POMDP under dynamic behavior is as shown in Fig. 4. It indicates that during warm up period, the cost of POMDP is relatively high. After that period, the cost of the POMDP with cluster number being 1, 2, and 3 is close to 1 even under behavior change. For the POMDP setting cluster number as 5, its cost starts to increase after the first behavior change due to failing to learn the behavior pattern correctly.

Conclusion and Future Work

This work proposes a POMDP-based approach to assist efficient decision-making for handling unforeseen events in VANETs. The POMDP model optimally queries information about uncertain events from neighboring vehicles while taking into account their malicious behavior. It better trades off the decision quality and the cost/delay incurred during information collection compared to the heuristic threshold-based schemes. This work also considers the realistic setting where the observation function representing the behavior of neighbors needs to be learned and proposes a learning algorithm that can handle dynamic scenarios where malicious vehicles change their behavior from time to time. Experiments demonstrate that the proposed POMDP approach can make decisions faster without sacrificing quality. The experiments under dynamic scenario verify that the learning algorithm can learn the observation function efficiently while still handling behavior change effectively.

As future work, we will conduct experiments with other kinds of sophisticated malicious behavior as well as include factors such as time, location, etc., while designing our POMDP model state. We will also explore the possibility of including dynamic reward functions.

Acknowledgments

This work is supported by the MOE AcRF Tier 1 funding (M4011261.020) awarded to Dr. Jie Zhang. Many thanks to Zehong Hu for the constructive discussion.

References

Chang, B.-J., and Kuo, S.-L. 2009. Markov chain trust model for trust-value analysis and key management in distributed multicast manets. *IEEE Transactions on Vehicular Technology* 58(4):1846–1863.

Ghavamzadeh, M.; Mannor, S.; Pineau, J.; Tamar, A.; et al. 2015. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning* 8(5-6):359–483.

Hartenstein, H., and Laberteaux, L. 2008. A tutorial survey on vehicular ad hoc networks. *IEEE Communications magazine* 46(6):164–171.

Irissappane, A. A.; Zhang, J.; Oliehoek, F. A.; and Dutta, P. S. 2015. Secure routing in wireless sensor networks via pomdps. In *Proceedings of the 24th International Joint Conferences on Artificial Intelligence (IJCAI)*, 2617–2623.

Irissappane, A. A.; Oliehoek, F. A.; and Zhang, J. 2014. A pomdp based approach to optimally select sellers in electronic marketplaces. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 1329–1336.

Irissappane, A. A.; Oliehoek, F. A.; and Zhang, J. 2016. A scalable framework to choose sellers in e-marketplaces using pomdps. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 158–164.

Jaulmes, R.; Pineau, J.; and Precup, D. 2005. Learning in non-stationary partially observable markov decision processes. In *Proceedings of European Conference on Machine Learning (ECML) Workshop on Reinforcement Learning in non-stationary environments*, 26–32.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101(1):99–134.

Kargl, F.; Papadimitratos, P.; Buttyan, L.; Müter, M.; Schoch, E.; Wiedersheim, B.; Thong, T.-V.; Calandriello, G.; Held, A.; Kung, A.; et al. 2008. Secure vehicular communication systems: implementation, performance, and research challenges. *IEEE Communications Magazine* 46(11):110–118.

Kurniawati, H.; Hsu, D.; and Lee, W. S. 2008. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Proceedings of Robotics: Science and Systems*, 65–72.

Li, W., and Song, H. 2016. Art: An attack-resistant trust management scheme for securing vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems* 17(4):960–969.

Li, Q.; Malip, A.; Martin, K. M.; Ng, S.-L.; and Zhang, J. 2012. A reputation-based announcement scheme for vanets. *IEEE Transactions on Vehicular Technology* 61(9):4095–4108.

Lin, R.; Kraus, S.; and Shavitt, Y. 2007. On the benefits of cheating by self-interested agents in vehicular networks. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 183.

Raya, M., and Hubaux, J.-P. 2007. Securing vehicular ad hoc networks. *Journal of computer security* 15(1):39–68.

Raya, M.; Papadimitratos, P.; Gligor, V. D.; and Hubaux, J.-P. 2008. On data-centric trust establishment in ephemeral ad hoc networks. In *Proceedings of the 27th Conference on Computer Communications (INFOCOM)*, 1238–1246.

Ross, S.; Chaib-draa, B.; and Pineau, J. 2008. Bayes-adaptive pomdps. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 1225–1232.

Sommer, C.; German, R.; and Dressler, F. 2011. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing* 10(1):3–15.

Sun, Y.; Lu, R.; Lin, X.; Shen, X.; and Su, J. 2010. An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications. *IEEE Transactions on Vehicular Technology* 59(7):3589–3603.

Wei, Z.; Tang, H.; Yu, F. R.; Wang, M.; and Mason, P. 2014. Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning. *IEEE Transactions on Vehicular Technology* 63(9):4647–4658.