Stuck in Traffic (SiT) Attacks:

A Framework for Identifying Stealthy Attacks that Cause Traffic Congestion

MINA GUIRGUIS
Dept. of Computer Science
Texas State University
msg@txstate.edu

GEORGE ATIA
Dept. of Electrical Engineering and Computer Science
University of Central Florida
george.atia@ucf.edu

***Abstract*—Recent advances in wireless technologies have enabled many new applications in Intelligent Transportation Systems (ITS) such as collision avoidance, cooperative driving, congestion avoidance, and traffic optimization. Due to the vulnerable nature of wireless communication against interference and intentional jamming, ITS face new challenges to ensure the reliability and the safety of the overall system. In this paper, we expose a class of stealthy attacks – Stuck in Traffic (SiT) attacks – that aim to cause congestion by exploiting how drivers make decisions based on smart traffic signs. An attacker mounting a SiT attack solves a Markov Decision Process problem to find optimal/suboptimal attack policies in which he/she interferes with a well-chosen subset of signals that are based on the state of the system. We apply approximate policy iteration algorithms to derive potent attack policies. We evaluate their performance on a number of systems and compare them to other attack policies including random, myopic and DoS attack policies. The generated policies, albeit suboptimal, are shown to significantly outperform other attack policies as they maximize the expected cumulative reward from the standpoint of the attacker.**

I. INTRODUCTION

In the area of Intelligent Transportation Systems (ITS), vehicle-to-vehicle and vehicle-to-infrastructure communications enable many potential applications such as collision avoidance, cooperative driving, congestion avoidance, and traffic optimization [1]–[5].

Due to the shared nature of the wireless channels used, the overall safety of the ITS is affected by interference and intentional jamming by adversaries. Jamming has been shown to cause severe effects that may cripple the whole system [6]–[9]. Previous incidents indicate the possibility of interfering with these communication mediums [10]. By placing jamming devices in vehicles and at critical transportation points (e.g., bridges, tunnels, cellular towers), an adversary can impact the overall traffic flow, exploit the adaptation of the drivers to make abrupt decisions that may cause accidents, or attempt to maximize their gain by preventing critical information from reaching a neighboring subset of vehicles [11], [12]. A much worse scenario may occur if a terrorist can create severe congestion in an area before detonating a bomb.

Many vehicles are already equipped with wireless connections to invoke traffic services. As drivers increase their reliance on wireless signals in making decisions, the absence or even the delay of these signals may have catastrophic effects due to the real-time constraints present in the system.

Drivers are typically faced with a decision making process whenever they encounter alternatives in choosing their routes. For example, should a driver use the upper or lower level when driving across George Washington Bridge? Should a driver use a highway or a local access road for a given short trip? The decisions made are not random, but are typically aided by traffic signs (e.g., reflecting the delay or the expected time to reach a particular point) and/or online map services (e.g., Google maps with traffic information). The goal is to reduce congestion as much as possible. It is known that traffic congestion is a significant problem that costs the US billions of dollars [13].

When wireless signals are used to communicate important information to drivers – through smart traffic signs and wireless transceivers in vehicles – jamming a subset of the signals may impact the overall traffic flow leading to unchecked safety conditions. In this paper, we expose a class of stealthy attacks – that we term Stuck in Traffic (SiT) attacks – that aim to cause congestion. By solving a Markov Decision Process (MDP) problem, an attacker mounting a SiT attack selects a subset of signals to interfere with. The choice of signals is based on the current state conditions of the system, taking into account the exposure risk the attacker is willing to take. Due to the exponential nature of the state space that describes the system, solving the MDP exactly is computationally prohibitive. Thus, we apply approximate policy iteration methods to identify suboptimal, yet efficient, attack policies.

The work in this paper relates to traffic safety and management applications, and security in V2V and V2I communication. Various safety scenarios, such as cooperative forward collision avoidance, intersection and lane-changes warning assistants, pre-crash sensing, and traffic optimization have been studied [1], [2], [14]. There has also been a large number of research studies that focused on the security of vehicular networks. Attacks on vehicular networks include stealthy attacks with low exposure aiming to hijack traffic [15], false-position data attacks on position-based routing networks [16], DoS attacks through jamming the communication channels, impersonation by using fake identities, and bogus information attacks wherein wrong data could be diffused in the network, for example to divert traffic from a given road. For a summary of various potential attacks on vehicular networks we refer the reader to [17] and references therein.

Contributions: The use of wireless technologies in various traffic safety applications is becoming the norm. Thus, it is important to expose potential security issues before deployment. In particular, we make the following contributions:

978-1-4673-6337-2/13/$31.00 ©2013 IEEE

- We provide a general framework for identifying stealthy attacks that reflect the best interest for an attacker: minimizing the cost while maximizing the damage.
- We expose SiT attacks that aim to cause traffic congestion through a proper choice of *which* signals to interfere with and *when*.
- In almost all the cases studied, we were able to identify attack policies that outperform myopic, random and DoS attack policies.

In Section II, we describe the framework developed to expose SiT attacks. We evaluate the impact of SiT attack policies in Section III and we conclude in Section IV.

II. A General Framework

In this section we present a framework that identifies stealthy SiT attack policies that create traffic congestion.

A. The Model

We consider a vehicular network that is composed of a set of segments and a decision point. A segment is a portion of an infrastructure (e.g., highway, bridge or a tunnel) that is controlled by one access point. As vehicles utilize a segment, they send wireless signals to the access point for the segment to get an estimate of its current load (e.g., number of vehicles). The access point reports its measurement back to a decision point to influence future incoming traffic. Each segment presents an alternative route to the driver. A decision point is a location at which drivers must make an "educated" decision on which segment to use (e.g., at highway entrance points and intersections). At each decision point, the load on each segment is presented to the driver. Figure 1 shows a diagram describing the setup.

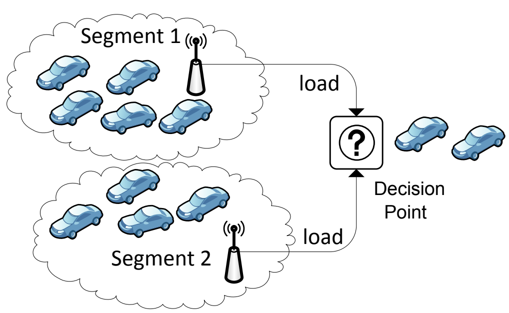

Fig. 1. A vehicular network with 2 segments and a decision point.

We consider a discrete-time model in which at each time step, new vehicles arrive at a decision point based on some arrival process. For simplicity, we assume an infinite population of vehicles for the arrival process. Based on the loads displayed, a driver picks an appropriate segment. Vehicles exit each segment based on its service rate.

Let λ_k denote the arrival rate at the decision point at time k, $\alpha_k(i), i = 1, \ldots, n$, denotes the admission ratio of vehicles on segment i at time k, and $\beta_k(i), i = 1, \ldots, n$, denotes the service rate for segment i at time k, where n is the total number of segments. Then the number of vehicles, $q_k(i)$ on segment i at time k is given by:

$$q_k(i) \quad = \quad q_{k-1}(i) + \alpha_{k-1}(i)\lambda_k - \beta_k(i). \tag{1}$$

Throughout the paper, we assume that the service rates are known and fixed. Depending on the traffic optimization policies, the admission ratio for each segment is determined based on the number of vehicles on all segments. Hence,

$$\alpha_k(i) = f\left(\hat{q}_{k-1}(1), \hat{q}_{k-1}(2), \ldots \hat{q}_{k-1}(n)\right) \tag{2}$$

where f is a traffic optimization function and $\hat{q}_k(i)$ is the estimate of the queue length of segment i at time k. The admission ratios can be chosen proportionally based on the number of vehicles on each segment, weighted by the service rate, or simply by picking the least loaded segment.

B. SiT Attacks

The goal of SiT attacks is to cause traffic congestion by jamming a subset of the wireless signals from the vehicles to the access points leading to incorrect estimates displayed for drivers, and consequently wrong decisions made by the drivers (e.g., choosing a congested segment). To reflect their stealthy nature, an attacker pays a price whenever he/she decides to jam a wireless signal. Clearly, if the cost of jamming is very high, SiT attacks would not jam any signal and if the cost of jamming is very low, SiT attacks would jam all the time (i.e., DoS attack). We are interested in identifying attacks policies that tradeoff damage and cost through deciding the proper attack action based on the current state of the system.

Let $s_k \in \mathcal{S}_k$ denote the state of the system at time k, where $\mathcal{S}_k$ is the state space at time k. The state of the system is the combination of the queue lengths $q_k(i)$ and $\alpha_k(i), i = 1, \ldots, n$. Based on the number of new arrivals and the admission ratios, the state is updated at the next time step.

The goal of the attacks is to unbalance the incoming traffic across segments by selectively choosing what signal(s) to attack at any state. Let $u_k \in \mathcal{U}_k$ denote the control action of the attacker at time k and $\mathcal{U}_k$ the control space, which depends on the state s_k. Note that the estimates $\hat{q}_k(i), i = 1, \ldots, n$, of the queue lengths are function of the attacker's control action u_k and the true queue lengths $q_k(i)$, i.e.,

$$\hat{q}_k(i) = h(q_k(i), u_k) \tag{3}$$

where h is some function, which for simplicity is assumed to be known to the attacker. Equations (1), (2) and (3) define the state dynamics. Note that from the attacker's standpoint, the state s_k consists not only of the queue lengths $q_k(i), i = 1, \ldots, n$, but also the admission ratios $\alpha_k(i), i = 1, \ldots, n$, since even for given values of the queue lengths, the attacker's course of action will change depending on the advertised admission ratios for the various segments.

The attacker's action at time k is obtained through a policy μ_k, which is defined as a mapping from the state space to the control space, i.e., $\mu_k : \mathcal{S}_k \to \mathcal{U}_k$.

Let $g(s, u, s')$ denote the reward obtained when the system evolves from state s to state s', under attack action u. The reward can be described by the following equation:

$$g(s, u, s') \quad = \quad d(s, u, s') - c(s, u, s') \tag{4}$$

where d is the damage function and c is the cost function of the attack action u. The damage d due to a SiT attack can be instantiated as the imbalance between different segments, weighted by their service rates, or the gap between the admission ratios reported to the drivers and the "true" admission ratios that *should have been reported*. Other forms of damage such as queueing delays and other factors that cause congestion could also be used. The cost of the attack is chosen to be proportional to the number of attacked vehicles to reflect the risk of exposure. An attack action becomes more appealing if it can cause higher damage with less cost.

The infinite horizon expected reward is given by

$$J(s_0, \mu_0, \mu_1, \ldots) = \mathsf{E}\left[\sum_{k=0}^{\infty} \gamma^k g(s_k, \mu_k(s_k), s_{k+1}) | s_0\right] \quad (5)$$

where s_0 is the initial system state and $0 < \gamma < 1$ is a discount factor. The discount factor ensures that immediate rewards have higher weights than future ones. Since the function $g(.)$ is bounded and $\gamma < 1$, the reward function (5) is well defined.

The attacker aims to maximize the cumulative expected discounted reward over time by choosing attack policies $\mu_0, \mu_1, \ldots$. Hence, the goal is to compute the solution to

$$J^*(s_0) = \max_{\mu_0, \mu_1, \ldots} J(s_0, \mu_0, \mu_1, \ldots). \quad (6)$$

The problem now falls within the class of infinite horizon problems with discounted reward. Hence, a stationary policy $\mu^*(.)$ which does not depend on k is optimal, and such policy can be obtained by solving the Bellman equation [18]:

$$J^*(s_0) = \max_{u \in \mathcal{U}(s_0)} \{\mathsf{E}[g(s_0, u, s_1)] + \gamma \sum_{s_1} p(s_1|s_0, u) J^*(s_1)\} \quad (7)$$

where $J^*(.)$ is the optimal value function. The first term on the R.H.S. represents the immediate stage reward in (4) and the second term is the future reward. The conditional probability $p(.)$ is the probability of a transition of the system to future state s_{k+1} from state s_k under attack action u, and hence the summation in the second term is over all possible future states, s_1 from s_0. Solving the fixed point equation above gives the optimal tradeoff between damage and cost.

Due to the large state space, solving the above equation using exact methods, such as exact policy iteration, may not be computationally feasible. Even for a small system with 2 segments with each potentially holding up to 100 vehicles, the size of the state space is 100×100, without accounting for the α which has a similar order of magnitude. Thus, we propose an Approximate Policy Iteration (API) method [19]–[21]. Before we describe the API methodology, we provide some brief background on exact policy iteration.

Exact Policy iteration consists of 2 steps: policy evaluation and policy improvement. In the policy evaluation step, we start with an initial policy μ. Then, we solve a system of linear equations to evaluate the cost function $J_\mu(s)$ starting from state s and using policy μ:

$$J_\mu(s) = \sum_{s'} p(s'|s, \mu(s)) \left(g(s, \mu(s), s') + \gamma J_\mu(s')\right) \quad (8)$$

where the summation is over the set of states s' that can be reached from state s and $g(s, \mu(s), s')$ is the reward obtained from the transition from s to s' under policy $\mu(s)$. In the policy improvement step, an improved policy $\bar{\mu}$ is generated according to the following equation:

$$\bar{\mu}(s) = \arg\max_{u \in \mathcal{U}(s)} \sum_{s'} p(s'|s, u) \left(g(s, u, s') + \gamma J_\mu(s')\right). \quad (9)$$

The improved policy is the one that maximizes the reward through selecting the best attack action u, from the set of actions $\mathcal{U}(s)$ available from state s. The improved policy $\bar{\mu}$ is then used as the new policy and a new iteration starts.

In the approximate policy iteration variant, we run Monte Carlo simulations to evaluate the current policy rather than solving the system of linear equations. We approximate $J_\mu(s)$ with a parametric representation $\tilde{J}_r(s)$:

$$\tilde{J}_r(s) = \sum_{j=1}^{M} r_j \phi_j(s) \quad (10)$$

where ϕ is a column of features (ϕ_j is the j-th entry), r is a row of weights, and M is the number of those features. The idea is to extract M features that characterize state s and approximate $J_\mu(s)$ by selecting r that solves a least square problem between the rewards obtained from the Monte Carlo simulations and the cross product of $r_j \phi_j(s)$. It is known that the linear combination of well chosen features can capture essential nonlinearities in the reward function [22], [23].

C. Feature Selection

Due to the approximation in our proposed methodology, we must rely on representative features to capture the fundamental characteristics of the state. We used the following features to approximate the value function $\tilde{J}_r(s)$ for state s with:

1) The number of vehicles on each segment.
2) The degree of imbalance between the number of vehicles on each segment (weighted by their departure rates when segments have different service rates).
3) The segment that is the least occupied.
4) The admission ratios reported to the drivers.
5) Difference between the true admission ratio and the one reported to drivers at the decision point.
6) How far the admission ratio is from the ideal one (e.g., 0.5 in case of two identical segments).
7) A flag to indicate if the two segments are empty.

III. Performance Evaluation

In this section we report on our evaluation of the approximate policy iteration (API) methods on a number of systems that are instantiated from the model described in Section II. In this paper we limit our evaluation to systems with two alternate segments. We fix the discount factor γ to 0.99.

A. System 1: Segments with equal service rates

Consider a system composed of 2 segments and 1 decision point. Vehicles arrive to the decision point per unit time based on the following probability distribution: 3 vehicles with probability 0.3, 8 vehicles with probability 0.6 and 30 vehicles with probability 0.1. Thus the average arrival rate is 8.7. We assume that the two segments are identical and each one has a maximum service rate of 5 vehicles per unit time. Based on the reported number of vehicles on each segment, the decision point reflects the admission ratio for each segment to balance traffic between the segments. In this system we assume that all the drivers follow the information displayed.

An attacker mounting a SiT attack jams a subset of the signals from vehicles to the access point. We assume that a SiT attack only affects 50% of the vehicles. Thus, the estimate $\hat{q}_k(i)$ in (3) becomes $\hat{q}_k(i) = \frac{1}{2} \times q_k(i)$ whenever the attacker decides to attack.[1] We take the cost function, c, to be a constant value, C_T, multiplied by the number of vehicles affected. We instantiate the damage function, d, to be the absolute difference between the number of vehicles on each segment since the attacker aims to unbalance the traffic between the two segments. At any state, the attacker can choose between the following actions: (1) Not attack with cost 0, (2) Attack half the vehicles on segment 1 with cost $C_T \times 0.5 \times q_k(1)$, or (3) Attack half the vehicles on segment 2 with cost $C_T \times 0.5 \times q_k(2)$.

We start our API method from 32 representative states that are chosen at 25 increments to give good coverage of the state space. Moreover, half these states reflect the true admission ratio while the other half have random admission ratios. We start with a random policy as a roll-out one. From each representative state, we run 50 independent trajectories and we compute the average reward across them. In each trajectory, we simulate the attack policy for 100 steps. In each iteration, a new policy is generated and we keep track of the weight vector r that produces the policy with the maximum reward.

Once the weight vector r is obtained, we compare between policies based on a completely different set of states that are generated at random. Hence, there is no intentional overlap between our training data and the ones we use for evaluation.

It is important to note that with API methods, there is no guarantee that the resulting policy is an improvement of the previous one as in the exact policy iteration method. Thus, we do not have a termination method except to run for a large number of iterations and to choose the best policy. We typically use between 100 and 1000 iterations.

Figure 2 shows the rewards obtained for different cost values C_T under different policies. Figure 2 (top) is for attack success rate 100%, Figure 2 (center) is for attack success rate 75%, and Figure 2 (bottom) is for attack success rate 50%. We compare our API method to a no-attack policy, a random attack, a DoS attack on one of the segments, and a myopic attack (in which only the immediate reward is used to select an action without regard to the future reward). We only show the interesting region based on the attack costs. If the cost of the attack is very low, API matches a DoS attack and if the cost is very high, API matches a no-attack policy. One can see that API tracks the best policies very well and in the majority of the cases, it provides the policy with the highest reward.

When looking at the percentages of different actions taken under different costs, our proposed policy smoothly adjusts the level of aggression based on the cost of the attack. For example, for System 1 with 75% attack success rate, the percentage of no-attack actions were 0%, 9%, 22%, 46%, 57%, and 71% for costs 0.125, 0.25, 0.5, 0.75,1 and 2, respectively.

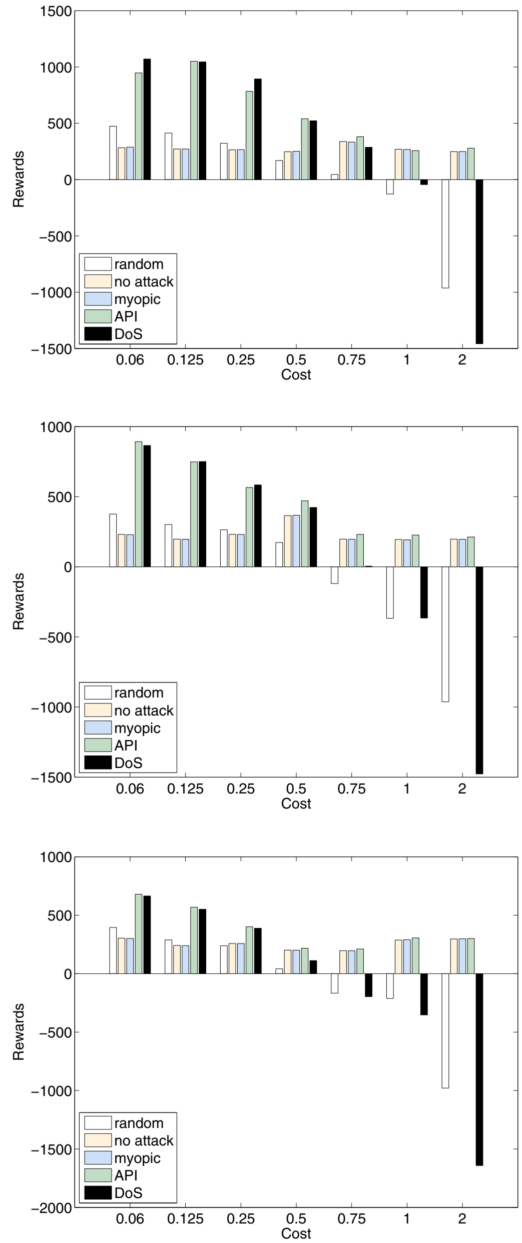

Fig. 2. Comparison between API, myopic, random, DoS and no attack for System 1 under different attack costs. Attack success rate is 100% (top), 75% (center) and 50% (bottom).

Notice also that the performance of the API method appears to improve as the degree of certainty in the attack success rate decreases. With an attack success rate of 50%, the API method was consistently better than all policies across all costs, whereas with higher attack success rates, the performance may be slightly less than some policies. This is because the API method accounts for the success rate when choosing actions that achieve a balance between immediate and future rewards.

[1]The percentage of affected vehicles depends on the duration of the jamming attack which the attacker seeks to minimize to remain stealthy.

B. System 2: Segments with unequal service rates

In this system, we consider segments that have different service rates. To make valid comparisons, we use the exact system as System 1 except that the maximum service rate for segment 1 and 2 are 4 and 6 vehicles per unit time, respectively. Both systems started with the same total number of vehicles weighted by their service rates. The initial admission ratio reflected the true admission ratio.

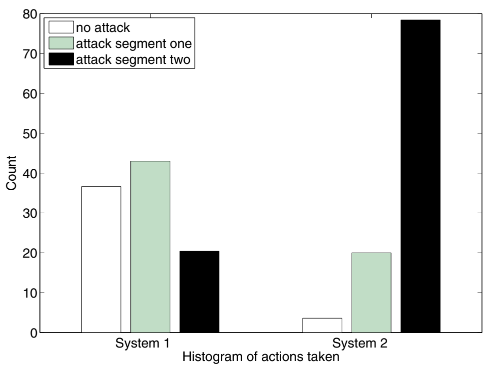

Fig. 3. Histogram of the actions taken under the same attack cost.

We have found that, under the same cost, attacks on systems that have segments with different service rates lead to higher rewards than attacks on ones that have segments with identical service rates. Figure 3 shows the average number of different actions taken by the attacker under cost 2, for both systems. One can see that for System 2, the best attack policy attacks either one of the segments more than 90% of the time when compared to System 1 in which the best attack policy attacks either one the segments only around 65% of the time.

Due to space constraints, we refer the reader to an extended version of this work where we present a more comprehensive evaluation including systems with a wider attack scope [24].

IV. Conclusions

In this paper, we developed a framework that is capable of exposing stealthy SiT attacks that aim to cause traffic congestion by selectively interfering with a subset of the signals from vehicles to the infrastructure. We have evaluated the generated attack policies and demonstrated their potency when compared to other policies such as myopic, random and DoS attacks. Unlike other policies, the proposed policy judiciously adapts to the system parameters (e.g., queue lengths, costs, and service rates) to select attack actions that balance between the current stage and future rewards. Moreover, we have shown that our proposed policy performs better as the degree of uncertainty in the system increases, making it appealing to adversaries that may not be confident of the exact impact of the attack. Furthermore, we have demonstrated that systems that employ segments with different service rates are more susceptible to attacks than those employing segments with similar service rates. To the best of our knowledge, this work is the first to look at the effect of jamming attacks through an MDP framework and to apply approximation techniques to identify stealthy attack policies that cause traffic congestion.

Acknowledgment

This work is supported by the NSF CNS grant #1149397.

References

[1] T. Elbatt, S. Goel, G. Holl, H. Krishnan, and J. Parikh, "Cooperative Collision Warning Using Dedicated Short Range Wireless Communications," in *Proceedings of VANET*, Los Angeles, CA, Sept. 2006.

[2] J. Misener, R. Sengupta, and H. Krishnan, "Cooperative Collision Warning: Enabling Crash Avoidance with Wireless Technology," in *Proceedings of 12th World Congress on ITS*, San Francisco, CA, 2005.

[3] G. Yan, W. Yang, M.C. Weigle, S. Olariu, and D. Rawat, "Cooperative Collision Warning through Mobility and Probability Prediction," in *IEEE Intelligent Vehicles Symposium (IV)*, San Diego, CA, June 2010.

[4] S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 74–82, 2006.

[5] M. Sichitiu and M. Kihl, "Inter-vehicle Communication Systems: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 10, 2008.

[6] L. Sang and A. Arora, "Capabilities of Low-power Wireless Jammers," in *Proceedings of INFOCOM*, Rio de Janeiro, Brazil, April 2009.

[7] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders, "Short Paper: Reactive Jamming in Wireless Networks: How Realistic is the Threat?," in *Proceedings of the ACM conf. on Wireless network security*, 2011.

[8] K. Pelechrinis, M. Iliofotou, and V. Krishnamurthy, "Denial of Service Attacks in Wireless Networks: The Case of Jammers," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, 2011.

[9] Y. Law, M. Palaniswami, L. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-Efficient Link-Layer Jamming Attacks gainst Wireless Sensor Network MAC Protocols," *ACM ToSN*, 2009.

[10] Diymedia, "FCC Enforcement News," http://www.diymedia.net/fccwatch/enfnews.htm.

[11] M. Raya and J. Hubaux, "Securing Vehicular Ad Hoc Networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[12] A. Aijaz, B. Bochow, F. Dötzer, A. Festag, M. Gerlach, R. Kroh, and T. Leinmüller, "Attacks on Inter Vehicle Communication Systems- An Analysis," in *Proceedings of the International Workshop on Intelligent Transportation (WIT)*, Hamburg, Germany, March 2006.

[13] Texas Transportation Institute, "2011 urban mobility report," http://tti.tamu.edu/documents/mobility-report-2011.pdf.

[14] O. Wolfson B. Xu and H. Ju Cho, "Monitoring neighboring vehicles for safety via v2v communication," in *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2011.

[15] M. Jakobsson, S. Wetzel, and B. Yener, "Stealth Attacks on Ad-hoc Wireless Networks," in *IEEE 58th Vehicular Technology Conference*, October 2003, vol. 3.

[16] T. Leinmüller, E. Schoch, F. Kargl, and C. Maihöfer, "Influence of Falsified Position Data on Geographic Ad-hoc Routing," in *Proceedings of the ESAS*, Berlin, Heidelberg, 2005.

[17] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications*, vol. 13, no. 5, 2006.

[18] R. Bellman, "A Markovian Decision Process," *Journal of Mathematics and Mechanics*, vol. 6, no. 4, 1957.

[19] L. Getoor and B. Taskar, "Introduction to Statistical Relational Learning," The MIT Press, 2007.

[20] D. Bertsekas, "Approximate Policy Iteration: A Survey and Some New Methods," *Journal of Control Theory and Applications*, 2011.

[21] R. Sutton and A. Barto, "Reinforcement Learning," MIT Press, 1988.

[22] D. Di Castro and S. Mannor, "Adaptive Bases for Reinforcement Learning," *Machine Learning and Knowledge Discovery in Databases*, pp. 312–327, 2010.

[23] H. Yu and D. Bertsekas, "Basis Function Adaptation Methods for Cost Approximation in MDP," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, Nashville, TN, April 2009.

[24] M. Guirguis and G. Atia, "Stuck in Traffic (SiT) Attacks: A Framework for Identifying Stealthy Attacks that Cause Traffic Congestion," Available at http://arxiv.org/abs/1210.5454, October 2012.