International Journal of Wireless Communication
ISSN: 2231-3559 & E-ISSN: 2231-3567, Volume 2, Issue 2, 2012, pp.-38-42.
Available online at http://www.bioinfo.in/contents.php?id=109

RECOGNITION OF MISBEHAVIORS IN VANET WITH INCORPORATED ROOT-CAUSE ANALYSIS

VELUKAR S.A.*, VISPUTE T.V., MAHAJAN G.V. AND GOSAWI P.R.

Department of Computer Science & Engineering, J.D.I.E.T, Yavatmal, Maharashtra, India
*Corresponding Author: Email- sarika.velukar@gmail.com

Received: March 15, 2012; Accepted: April 12, 2012

Abstract- In this paper we have introduced a novel format for Misbehavior detection schemes. Misbehavior detection schemes form an basic part of disobeying node ejection in vehicular ad hoc networks (VANETs). A misbehaving node can send messages correspondent to an incident that either has not occurred, or incorrect information corresponding to an actual incident, or both, causing applications to breakdown. When misbehavior is identified, it is vital to extort the root cause of the observed misbehavior This paper uses the Post-Crash Notification (PCN) application to illustrate the basic considerations and the key factors affecting the dependability recital of such schemes. The basic cause-tree approach is used effectively to jointly achieve misbehavior detection as well as identification of its root-cause and approach is illustrated. The approach is to first assemble a cause-tree, and then use successive logical reduction to arrive at a decision indicating the root-cause of the misbehavior. Misbehavior detection delay can be thought of as inversely correlated to the probability of detecting misbehavior by a vehicle. In this paper we will see this prospect and the probability of incorrectly declaring misbehavior as the performance metrics. The dependence of this reliability performance on the micro-mobility model of the vehicles is studied.
Keywords- VANET, MDS, PCN, basic cause-tree approach

Citation: Velukar S.A., et al. (2012) Recognition of Misbehaviors in Vanet with Incorporated Root-Cause Analysis. International Journal of Wireless Communication, ISSN: 2231-3559 & E-ISSN: 2231-3567, Volume 2, Issue 2, 2012, pp.-38-42.

Introduction

We are witnessing an inimitable junction of Vehicular Ad-hoc Networks (VANET) and Intelligent Transportation Systems (ITS) which is on the edge to bring about a innovatory leap by making our roadways and streets safer and the driving experience more enjoyable [1]. Working with the fielded ITS infrastructure, VANET is expected to boost the consciousness of the traveling public by aggregating, propagating and disseminating up-to-the-minute information about impending traffic-related measures. The main aim of this technology is to give drivers more comfortable and more secure driving experience. Based on automatic information exchange between cars and infrastructures, the drivers could know the road conditions or the information about the parking lots immediately. A Vehicular Ad Hoc Network, or VANET, is a special kind of MANET in which the mobile nodes are all vehicles equipped with an On-Board Unit (OBU) that enable them to send and receive messages from and to the other nodes in the network. In addition to communication among the vehicles, a VANET might also interface with communication points provided by onroad infrastructure.

The V2V applications broadcast messages that contain the type of the message and possibly other application specific information. Each message also contains some authentication information [9] to help the receivers validate the authenticity of the information. In particular, appended to each message is (a) digital signature on the message using the private key of the sending entity, (b) public key of the sending entity, and (c) a certificate on the public key issued by a trusted third party, the Certificate Authority (CA). Before passing it on to the relevant application layer, the digital signature is required to verify at security layer of receiver. At security layer a simple credential-validity check is performed by the receiver to confirm whether the certificate of the sender is

in the copy of the Certificate Revocation List (CRL) available. A CRL contains a list of known misbehaving certificate identities [3], so that if the certificate id of the sending entity appears in the CRL, the message could be discarded. The receiver would have downloaded the CRL during some of its last interaction with the infrastructure, which could be in the form of a Road Side Entity (RSE) connected to the CA. Owing to the sparse infrastructure presence in VANETs, detection of misbehaving vehicles (certificates) inevitably requires feedback from the participating entities. A participating vehicle runs some misbehavior detection scheme (MDS) to detect a misbehavior, which is then reported to the CA. The CA accumulates some number of reports of misbehavior against any certificate before revoking the certificate and populating the corresponding CRL[7]. Any vehicle requesting for the CRLs then receives the new information, leading to eviction of newly detected misbehaving vehicles. The final security performance thus depends on the detection delay (DD), the reporting delay (RD), and the eviction delay (ED).
In this paper, we focus only on the design of misbehavior detection schemes. In this paper we introduce an MDS and analyze the dependence of its reliability performance on the micro-mobility model of the vehicles and its parameter estimation. VANET provides with the safety application and one of its safety application in which the VANET is used to identify conditions that could potentially endanger the driver's safety [8]. Safety application used is Post Crash Notification (PCN) application .In this paper, we focus on the Post Crash Notification (PCN) application. The PCN application informs the driver when there is a crashed vehicle ahead on the same roadway. Post Crash Notification is in which vehicle involved in a crash broadcasts a PCN alert to the vehicles in its vicinity to inform them of the existence and the location of a crash, thus enabling them to take evasive action. A PCN alert is normally sent by a car involved in a crash. The PCN alert contains the position of the crashed vehicle, heading, and vehicle status [2]. A malicious vehicle could send out wrong PCN alerts with false position information even if there is no crash. On the other hand, it could be the case that the crashed vehicle's sensors are faulty so that they are sending out incorrect location information. The action taken on detecting misbehavior may vary with the severity of the potential consequences of the root-cause of the misbehavior. For example, consider the case when nodes have to rejoin the V2V network after they were revoked due to the broadcast of incorrect information either due to malicious intent or sensor malfunction. Hence understanding the nature or root of the misbehavior is an important step in determining the post misbehavior detection scenarios. The misbehavior detection schemes (MDSs) could thus be required to not only detect misbehavior, but also identify the root-cause of the misbehavior. This could be the case when the action taken on detecting misbehavior may vary with the severity of the potential consequences of the root-cause of the misbehavior. The paper introduces the MDS for the PCN application that can identify the root-cause of the misbehavior. The approach is to first construct a cause-tree, and then use successive logical reduction to arrive at a decision which indicates the root-cause of the misbehavior. The rest of the paper is organized as follows.

Misbehavior Detection Scheme

We now develop the misbehavior detection scheme for Post Crash Notification alerts. An OBU that needs to implement an MDS has three sources of information about the system that might help it in the construction of the MDS. The three sources of information are primary, secondary information and information form collaboration[8].The MDS proposed by Ghosh et al. [3] for the PCN application uses precomputed descriptions of expected driver behavior to compute an expected driver trajectory in the presence of a crash, and then compares this expected trajectory to the actual path followed by the driver. If the deviation is larger than a certain threshold, misbehavior is declared. The development of scheme presents the generic basic considerations first, and then specializes them to infer the root-cause by using logical reduction.

Overview

The Warning application (PCN) alerts the traffic with the disabled vehicle that is stuck in or near traffic lanes to enable drivers to choose other lanes if possible. The introduced approach relies on observing the driver's behavior after receiving an alert. Based on other neighborhood or visual inputs, the driver can determine if there is really a crash or if the alert is false. If the driver finds the alert to be true, he/she will take necessary actions and the car will move according to the crash-modulated mobility model defined above until it crosses the crash site. On the other hand, if the driver finds the alert to be false, he/she will continue to move following the free-flow mobility model since there is no crash. In the introduced MDS, the On-Board Unit (OBU) of a vehicle P raises a PCN alert, which is received by other vehicles in the vicinity. Consider a vehicle Q getting the PCN alert. The MDS in succession in the OBU of Q needs to decide if the notification received is true or false. The scheme is based on the OBU of Q observing its driver's behavior for some time after the notification is received, comparing it with some anticipated behavior of the driver, and identifying different root-causes based on the observed deviations between the two. The movement of vehicle in the absence of any notification is assumed to follow some freeflow mobility model, and its movement after receiving an notification is assumed to follow some crash-modulated mobility model. The position of the receiving vehicle when it receives the notification is nominated as position 0. This is the origin of reference for all other position values. With reference to this origin, we use the following two notations to define two other positions:

Dp : the position of the crash as reported in the PCN alert

Dc : the actual position of the crash if any

Thus, Dp is where the receiving vehicle thinks the crash is, whereas Dc is the actual position of the crash if any. Note that the OBU does not know Dp. The OBU observes the driver's activities after receiving an alert till some position X after Dp. This is required since the OBU is not aware of Dc. If the driver finds the alert to be true, he/she will take necessary actions and the vehicle will move according to the crash-modulated mobility model until Dc and from then on it will trail the free-flow model till X. On the other hand, if the driver finds the alert to be false, it is expected that the driver will continue to stick to the free-flow mobility model until X. The location of vehicle information containing lane number and distance from the origin is sensed by the OBU at predefined number of sampling divided till distance X.The trajectory of the vehicle is comprised of progression of location information.We use the following two notations for denoting the expected trajectories.

- Gexp[u,v]: the expected crash-modulated trajectory from position u to position v.
- Fexp[u,v]: the expected free-flow trajectory from position u to position v.

Depanding on the initial lane the vehicle is in when the alert is received, and the lane the crash is reported from, an expected trajectory Gexp[0,Dp] of the vehicle following th ecrash-modulated mobility model is calculated. Here 0 represents the position where the vehicle receives the alert that is origin. As Dc is not known to the OBU and the driver chooses to go through its free-flow behavior after Dc, an expected free-flow trajectory is also calculated from Dp to X .The MDS performs comparision between the expected trajectory and the actual sensed trajectory. An MDS that is not required to deduce the root-cause would only use the distance [4] between the expected trajectory and the actual sensed trajectory. However, deducing the root-cause will require supplementary effort. We next describe the use of the expected and sensed trajectories to arrive at the root-cause.

Mutual assumption of misbehavior and its root-cause

Let us assume that the speed of a vehicle does not change as a function of lane or time. Let xt denote the actual lane number of the vehicle at the tth sample point. Similarly, let xt denote the expected lane number of the vehicle at the tth sample point. Then the deviation d between two trajectories, expected and actual, over t sample points starting from position 0 is obtained using the following [5]. For this equation the limit is from t=0 to t.

$$d = \sum[(x_t - x'_t)^2]$$

The difference between the expected trajectories and the actual trajectories are calculated for different distances and the following deviations are obtained:

- dG (0,Dp): deviation between the actual trajectory and G exp [0,Dp]
- dF(0,Dp): deviation between the actual trajectory and Pexp[o,Dp]
- dF(Dr,X): deviation between the actual trajectory and Pexp[Dp,X]

Depending on these variations, the varient misbehavior cases are identified as described bellow. The misbehavior detection scheme is graphically represented as a cause-tree as shown in Fig. 1[6].

The leaf nodes how the different cases possible and the corresponding deviation values are given for each d defined in the earlier section. The parameters Î1 and Î2 represent thresholds which denotes how close an actual trajectory is to the expected crash-modulated trajectory and the expected freeflow trajectory respectively. A suitable choice of Î1 and Î2 can be used to recognise the variours misbehavior cases. The following table [3] shows the different possible cases that can arise depending on whether the alert is true or false and whether it is detected correctly or not.

S	Misbehavior	Legitimate
Detected(dist>Î)	True Positive	False Positive
Not Detected(dist<Î)	False Negative	True Negative

The explaination of each of these cases are as follows.

Case 1

True alert with correct position information: In this case, the driver follows the crash-modulated trajectory till Dr and then changes to the free-flow trajectory.

Case 2

True alert with false position information: In this case, the driver follows the crash-modulated trajectory for some time. But the crash-modulated trajectory depends on the position information and therefore a false position information will affect it. The following subcases are possible:

Case 2(a)

Dc > 0 and Dc < Dp: In this case, the driver will come across the crash site earlier than expected. The MDS will persist to calculate the deviation till Dr. The actual trajectory of the driver will follow the crash-modulated trajectory till Da, and then it will follow the free-flow trajectory .

Case 2(b)

Dc > 0 and Dc > Dp: In this case the driver, on reaching the crash site, will not find a crash as the actual crash is farther away. The driver moves with the crash-modulated trajectory till Dr. However, after Dr , the behavior of the driver is uncertain. The driver, on not seeing the crash at Dr , but expecting a crash as a PCN alert is received, is expected to deviate from his/her free-flow behavior.

Case 2(c)

Dc < 0: Here an assumption is made that if a crash has actually taken place somewhere before then the driver has seen it in the recent past. The driver will continue to move with the free-flow trajectory on receiving an notification and knowing it is for the crash just passed back.

Case 3

False alert: In this case, since no crash has taken place, the driving conditions that the driver faces do not change since no crash has taken place. Thus the driver will not have to adjust to any changed driving condition and hence would mostly continue with its free-flow behavior. So this case is similar to Case 2(c) above with similar checking conditions.

The final MDS for PCN alerts is derived directly from the tree shown in Fig. 1[6]. The pseudocode is shown in Algorithm 1[6]. Note that we can separate Cases 1 and 2(b) from Cases 2(a), 2 (c), and 3 using dG (0,Dp) only first, and then further identify the individual cases using dF (0,Dp) and dF (Dp,Xs). This is reflected in the algorithm shown.

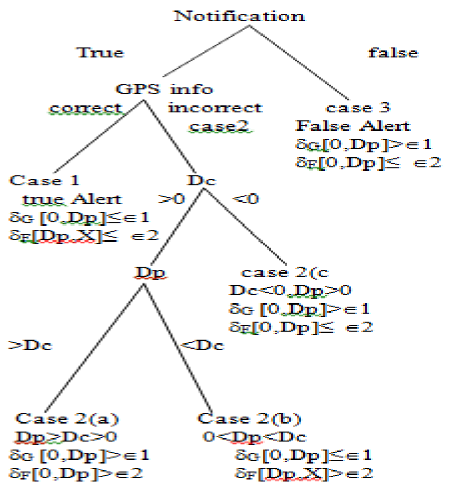

Fig. 1- Tree showing the different checking conditions for the different cases

Algorithm 1. Algorithm for misbehavior detection

1: Calculate dG (0,Dp); dF (0,Dp)
2: if dG (0,Dp) > Î1
3: if dF (0,Dp) > Î2
4: Report Case 2(a)
5: else
6: Report Case 2(c) or Case 3
7: end if
8: else
9: Calculate dF (Dp,X)
10: if dF (Dp,X) > Î2
11: Report Case 2(b)
12: else
13: Report Case 1
14: end if
15: end if

The probability of not detecting a misbehavior of any type depends crucially on the thresholds Î1 and Î2. The thresholds should be chosen judiciously in order to make this probability low. Figure 1 tree showing the different checking conditions for the different cases.

Result and output

Several experiments were done to evaluate the performance of the MDS approach. An estimate of the projected movement of the vehicle under a crash is first calculated. This is evaluated by generating a very large number of paths by making use of the M matrix and then averaging the vehicle location at every time slot. Thus, the expected crash-modulated trajectory gives the location of the vehicle at τ time slots, averaged over a very large number of generated crash-modulated trajectories. A large number N of sample freeflow trajectories are then created and made comparision with a time-slot-by-time-slot basis with the projected trajectory based on the distance metric.

TheMDS method is simulated with a vehicle system. The vehicle following a random path generates crash alerts randomly. Thevehicle at the back follows a mobility model defined by P. When tah alert is received, the vehicle continues with the free-flow model if it come to the result that the alert is false, otherwise it follows the crash-modulated model defined by M. The difference between the expected crash-modulated trajectory and actual path is recorded and the probability of not detecting a misbehavior is calculated.

The MDS model used

The probable results we present are for an n-lane highway where each lane has a selected average speed. For the free-flow model, the lane number of the vehicle at the ends of slots is approximated by OBU by a Markov chain [6] with an n transition probability matrix P. Over time ,the OBU estimates the parameters of the Markovian transition probability matrix P. The (i,j)th entry of P gives the projected probability that the driver, if currently on lane i, will change to lane j in the subsequent time slot. The OBU assumed that if crash occurs then the movement of the vehicles involved at the place of crash is to be governed by the transition probability matrix T.For example, for two lane (1 and 2), if the crash occurred in the first lane, T would be of the form:

T=

In this example, a vehicle on lane 1 will always move to lane 2 at the crash site because the crash is on lane 1.

As the vehicle reaches to the crash site, vehicle's movement transitions from the free-flow model dictated by P to that given by T. During this transition, the movement of the vehicle can be modeled by a modulated transition probability matrix M of the form

$M = (1 - \alpha)P + \alpha T$,

where $0 < \alpha < 1$ and the value of α increases as the distance to the crash site of the receiving vehicle decreases.

we have implemented the Misbehavior Detection Scheme in VANET wtih Post Crash Notificatin application. Whenever the crash take place the PCN alert (Post Crash Notificatin) is notified to the near by vehicles, which are in the range of the vehicle. The following figure 2 shows the implementation of the MDS scheme.

The fig.2 has 10 nodes which resembles vehicles. When the PCN alert is received by the near by vehicle that vehicle will perform checking of different conditions in the MDS scheme. The MDS decide whether the PCN alert is true or false. Then the Gps information is checked whether it is correct or incorrect.

If the received information is found to be correct then scheme evaluates the alert as true alert and then follows the crash-modulated trajectory till the crash position and then it free-flow trajectory. If the received information is found to be incorrect then the MDS scheme evaluates the alert as true alert with false information and then performs the cases as explain earlier.If the alert received by the vehicle is false alert then it follows free-flow trajectory.

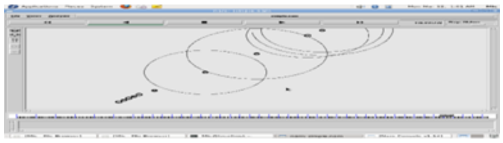

Fig. 2- output showing range of node (vehicle)

The nodes are shown by black spots, here we have created 10 nodes (vehicle). The circles shown surrounding the nodes indicates the range of particular nodes. The node can send message in that range of segment only. If the nearby vehicle is not in the range then the message sent is lost

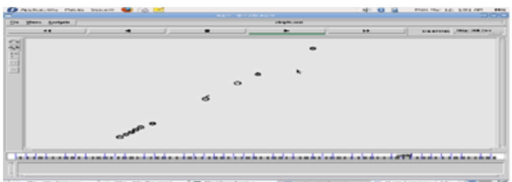

Fig. 3- output showing the movement of node (vehicle) and alert sending

The fig.3 shows the result image in which nodes resembles vehicles. The vehicles, moving in the vicinity of nodes, if crash takes place then it sends the PCN alert implementing MDS as explain earlier. If the data packets of notification is not received by nodes as not being in the range then data packet loss occurs. This lost of data packets is shown by dark dashed lines in the fig.3.If again node come in the range then packets are transmmited. As here we are implementing only alerting nodes of crash with PCN alert, recovery from crash is the next step of us.

Conclusion

In this paper, we have presented and evaluated a misbehavior detection scheme for PCN application. The results indicate that the scheme performs well in detecting misbehaviors while reducing the chance of false positives and false negatives. It is to be taken into consideration that the PCN alert raised will be received by multiple vehicles in the vicinity, and more than one of these receiving vehicles may use an MDS in their OBUs to detect misbehavior. To improve the detection rate one possible way can be for nearby vehicles to collaborate and exchange their results. Design of collaborative schemes is a challenging problem. The problem of inferring the actual location of the crash appears to be challenging.

Acknowledgement

We express sincere gratitude to our guide for providing their valuable guidance and necessary facilities needed for the successful completion of this paper throughout. . Last but not least, we thank our parents for their support and thank all friends and well-wishers who were a constant source of inspiration.

References

[1] Chen W., Cai S. and Inc T.(2005) *IEEE Communications Magazine*, 43(4), 100-107.

[2] Crash Avoidance Metric Partnership (CAMP) (2004) *Vehicle Safety Communications Project*.

[3] Ghosh M., Verghese A., Kherani A. and Gupta A. (2009) *IEEE Wireless Communication and Networking Conference*.

[4] *http://www.onstar.com*.

[5] Raya M., Papadimitratos P., Aad I., Jungels D., Hubaux J.P. (2007) *IEEE Journal on Selected Areas in Communications*, 25, 1557-1568.

[6] Mainak Ghosh, Anitha Varghese, Arobinda Gupta, Arzad A. Kherani , Skanda N. Muthaiah (2010) *Detecting misbehaviors in VANET with integrated root-cause analysis*, 778-790.

[7] Rao A., Sangwan A., Kherani A., Varghese A. , Bellur B., Shorey R. (2007) *Mobile Networking for Vehicular Environments*, 127-132.

[8] Yin J., ElBatt T., Yeung G., Ryu B., Habermas S., Krishnan H. and Talty T. (2004) *The 1st ACM international workshop on Vehicular ad hoc networks*, 1-9.

[9] IEEE Trial-Use Standard for Wireless Access in Vehicular Environments, *Security Services for Applications and Management Messages, IEEE.*