LAVI: A Location Aware Virtual Infrastructure for VANETs

Rohan B. Golatkar†, Yu Chen†, Wei-Shinn Ku‡
†*Dept. of Electrical and Computing Engineering, SUNY-Binghamton, Binghamton, NY 13902, USA*
‡*Dept. of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, USA*

***Abstract*—Without requiring pre-fixed infrastructure, Vehicular Ad Hoc networks (VANETs) allow drivers to exchange information and access variant services in real-time manner. While achieving significant flexibility and convenience, dynamic network topology is also the source of challenges that invalidate most of well-designed protocols. To address this problem, in this paper, we propose a novel scheme called Location Aware Virtual Infrastructure (LAVI). The fundamental rationale of the LAVI scheme is based on an observation: although each individual vehicle will not stay at a location for an extended period of time, statistically there are some vehicles available as long as the density of vehicles is reasonably high. LAVI creates a virtually stable infrastructure layer on top of the dynamic physical topology. Through extensive experimental study, we have verified the effectiveness of our LAVI scheme.**

***Keywords*-Virtual infrastructure, VANETs, Location information.**

I. Introduction

Vehicular Ad hoc Networks (VANETs) allow vehicles to communicate through peers around them without the help of a pre-existing infrastructure. VANETs make communication possible in places where there is no infrastructure available or existing communication facilities have been destroyed. For example, it is critical to form an ad hoc network in battlefield operations, real-time traffic information exchange among vehicles, and disaster relief efforts [7], [15].

Although infrastructure independence makes VANETs flexible, it invalidates some existing protocols that work well in normal networks where fixed infrastructure is available, such as the Internet. Alternative approaches have been proposed by the research community to address challenges including robust routing, multicasting, quality of service, media accessing, energy-economic design, network management, cross-layer protocols, and security issues [2], [9], [14].

However, the activities of mobile nodes may provide some helpful information. Lets take the vehicles on the road everyday as an example. In a city or a community, most of the vehicles travel among limited destinations repeatedly, which are determined by the activity of drivers. Each single movement is not predictable. However, statistically the set of nodes of a VANET in an urban area are stable and the areas they may appear are limited.

For instance, an individual living in a city would be repeatedly driving among places including their home, working place, local grocery stores, doctors office, schools to pick up the kids, and so on. In addition, these limited destinations also imply that the peers a vehicle meets on the road are not all strangers. Therefore, it is feasible for vehicles to determine whether peers are willing to cooperate. Here, the cooperation is defined in terms of cooperating to carry out communication tasks such as helping to store/forward packets.

We propose a novel *location aware virtual infrastructure* (LAVI) scheme that tries to address the fundamental problem of dynamic network topology. LAVI is able to provide a stable virtual infrastructure statistically on top of a collection of mobile nodes. The rationale results from the above observation regarding the daily activities of the owner of a vehicle. Since it is the same set of peers that a node deals with, a low cost simple recognition memory [1] can evolutionarily establish an efficient cooperation relationship among nodes. The principles of recognition memory and application algorithm will be discussed in detail in section 3.

For this purpose, we assume that each vehicle has the capability to obtain its location information. The location information is critical for four essential functionalities to establish LAVI. 1) It is used to make routing decisions when a node needs to forward packets; 2) When a vehicle moves a certain distance, if it is leaving an area, it needs to hand off routing information to another vehicle; 3) Based on the location and moving direction, a vehicle can decide whether it is the best candidate to take over the responsibility; and 4) In case there is no vehicle in a certain area, the location information is useful to update the network topology.

The rest of the paper is organized as follows: Section 2 gives a brief review of related works. Section 3 introduces the rationale, methodology and architecture of our LAVI protocol. Section 4 presents the simulation results and performance discussion. Section 5 presents some important discussion. Section 6 concludes the paper.

II. Related Work

VANETs have been studied extensively in recent years. A plethora of efforts have been reported to meet the challenges resulting from the lack of a pre-existing infrastructure [2], [9], [15], [16]. Researchers have addressed the problem from different perspectives to serve different purposes, such as routing [9], security [5], [14], Quality of Service (QoS) [4], etc. Since our work aims to provide a stable infrastructure in a dynamic wireless environment, this section gives a brief discussion about closely related research work.

There are reported efforts trying to take advantage of location services to implement multicast routing protocols for VANETs [8], [17]. However, the location information is merely an approach for routing or node finding. None of these solutions considered constructing a virtual infrastructure based on location services.

LAVI is also related to the idea of cluster based routing protocols in which nodes in a network form clusters. In a cluster, a node functions as the head which takes the responsibility of routing. In [13], an inter-cluster communication approach has been implemented via gateway nodes. Gateway nodes are the nodes that belong to two or more clusters. Each cluster has one cluster head, which acts as a base station for that cluster just like base station or a router in wired networks. In order to route the packets it uses a path discovery process just like the Dynamic Source Routing (DSR) protocol or the Ad hoc On Demand Distance Vector (AODV) protocols [10], [12].

In our LAVI scheme, the entire region is divided into multiple small units called cells. Each cell has its own base station. When this node moves to another cell it hands over its responsibility as a cell leader to another qualified node. After the design was done, we found this idea has been proposed in [11]. However, SOLONet was designed for MANETs and our protocol is designed specifically for VANETs. There is less or no randomness in the movements of nodes as compared to MANETs. The major difference between SOLONet and LAVI lies in the cell leader switching algorithm. When the current virtual infrastructure node (VIN) is moving to another cell it hands over its responsibility to a recognized peer based on the recognition memory that we are using in addition to the time to live value for each node. Use of content based recognition memory to maintain the database of co-operative nodes makes our routing protocol more robust and secure. There are also some other minute differences in the design of our LAVI and SOLONet.

The initial idea of LAVI was proposed in 2010 and the basic rationale has been illustrated in a short position paper [3]. This paper reports the complete design of the scheme and simulation experimental results.

III. Location Aware Virtual Infrastructure

This section introduces the rationale of the LAVI scheme first, followed by the discussion of a mechanism for VIN switching. Then, a more robust lightweight VIN switching algorithm based on recognition memory is presented, which improves the success rate of VIN switching by enabling mobile nodes to tell cooperators from defectors.

A. Rationale of LAVI Scheme

The essential purpose of the LAVI scheme is to provide a stable logic infrastructure for VANETs that is capable of tolerating the network topology dynamics due to the nodes mobility. Therefore, the rationale is to construct an overlay network that screens the node mobility as long as the mobility will not break the logical connections in the overlay layer.

The area covered by the network is divided into multiple hexagonal cells. With the help of a GPS system, each vehicle is aware of its location inside a cell as well as the range of the cells. In each cell, there is one and only one vehicle plays the role of contacting point. Such a node is a VIN that maintains the logic connections between the cells. When a vehicle enters a cell, it contacts the local VIN to register as a member of this cell. The communication with other peers goes through the VIN if the peer is not in the same cell. Similarly, while a vehicle is moving away from a cell, it needs to de-register from the VIN. This procedure is similar to the hand on/hand off operation in cell phone architecture.

If a cell is empty, the new mobile node cannot register to a VIN when it enters. Under such a scenario, the new node will automatically assume itself as the VIN of this cell and start to contact the VINs of the cells around it. This implies that a new virtual node is established in the virtual infrastructure. When the VIN node moves close to the edge of the cell, it hands the duty to one of the registered vehicles. Therefore, the network topology will not change when a VIN node is moving away from a cell if there is another vehicle in the range that can serve as the VIN node.

Effectively, a semi-dynamic network topology is achieved as the instability resulted from the mobility has been hidden. Intuitively speaking, a stable network infrastructure is available if the density of mobile nodes is reasonably high. It is not a challenge to maintain a certain node density considering most of the typical applications of VANETs. For instance, in a battlefield it is reasonable to expect a certain amount of soldiers and/or vehicles; in the case of disaster recovery, the density of vehicles and personnel involved would not be too low in certain areas.

B. Preliminary VIN Switching Mechanism

The VIN of each cell maintains a simple data structure called register table (RT), which records the location information of all mobile nodes in the cell. The contents of the RT are updated periodically to track the nodes in a real-time manner. When a mobile node enters a cell and registers with the VIN, a new entry is created; this entry will be removed when the node leaves the cell. Figure 1 shows an example of how VINs update their RTs when node NA moves from Cell#1 to Cell#0. Note that the VIN also registered itself in the RT with its original node ID.

In the register table, one of the most important items is the TTL (time to leave) that tells how long a mobile node will be in the cell. Normally, it is difficult to evaluate TTL precisely due to the lack of information to predict the nodes activity. However, some constraints in the physical world may be helpful. A vehicle moving along the street will not

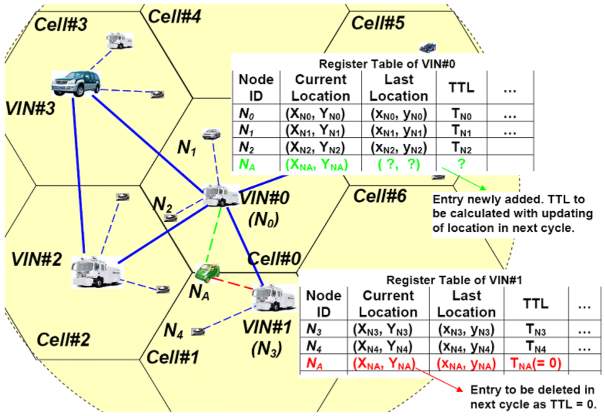


Figure 1. RT update in VINs.

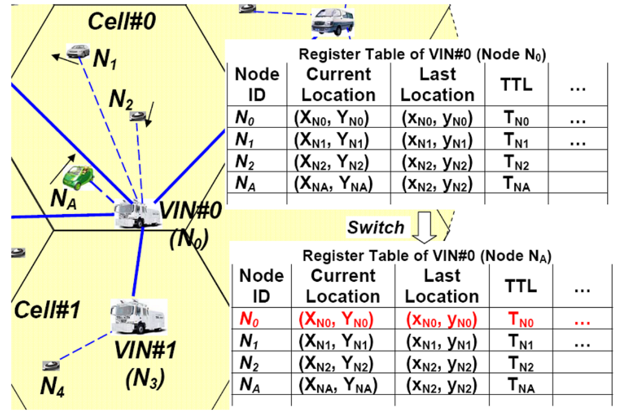


Figure 2. Preliminary VIN switching procedure.

change its speed and/or direction randomly. Instead, it has to obey the traffic rules and move along the street. Hence, by updating the position of the nodes periodically, the VIN can calculate the moving speed and direction of a vehicle. With the knowledge of location and geographic condition of a vehicle, we can roughly tell how long a node will be in the cell.

To minimize the overhead, less VIN switching operation is desired. When a VIN node is leaving the cell, it looks in the register table to find the node with the largest TTL value. Then the new VIN node will obtain the register table and other data from the leaving VIN. If the VIN is the only node in that cell, it needs to notify the VINs in the neighboring cells to update the network topology since one connection is disappearing.

It is highly desired that the VIN switching operation is transparent to upper layer applications. As long as there is a successor in the cell, there should be no change observed from the perspective of network topology, routing, QoS, etc. To achieve this goal, the LAVI architecture is designed as a logic network on top of the physical VANETs. Each VIN matches a mobile nodes ID when it plays the role of a routing point. The binding of a VIN ID to the mobile node ID is transparent to upper layers.

Figure 2 illustrates the preliminary VIN switching procedure. Based on the current location and previous location, the VIN determines the direction and speed of each node in the cell. This information in turn tells the VIN that TNA is the largest among all TTLs currently in RT. Then NA is selected to be the new VIN and the RT is transferred to it. In the new RT, TN0 is small, and this entry will be removed once N0 leaves the cell. To VIN#1, N0 is nothing more than a newly arrival common node and is registered normally.

C. Recognition Memory based VIN Switching

There is no guarantee that the vehicle with the largest TTL value is willing to cooperate. It would be even worse if such a node is malicious, which will mistreat the data packets forwarded through it (for example, reading/modifying contents he/she is not authorized to access, or simply dropping the packets). Techniques to detect malicious nodes are beyond the scope of this paper.

Recent achievements in understanding the decision process revealed that simple strategies can lead to successful and adaptive behaviors [1]. In particular, a memory of previous cooperators and defectors can effectively promote the evolution of cooperation [1]. With the observation that the peers a vehicle meets on the road are not all strangers, we propose a novel scheme to optimize the VIN switching operation based on the evolutionary cooperation model using recognition memory.

Each mobile node is equipped with a small recognition memory that can recognize a previously encountered peer. Associated with the recognition is an indicator telling whether this encountered peer was cooperative or not. The recognition memory does not need much space to keep a complicated historical cooperation record, only a flag bit is set.

The cost of recognition memory is low. Two 128-bit Bloom filters are adopted to record the cooperators (C-Filter) and defectors (D-Filter) respectively. The mobile node IDs are the inputs for both filters. When a VIN finds the node with the longest TTL value in the register table, it verifies that there is a record of cooperation. Here we call it a hit if a node ID matches a record in the C-Filter or the D-Filter. If a hit is obtained in the C-Filter, it is ideal to transfer the role of VIN to that node since it was a cooperator before. If a hit in D-Filter is observed, it implies that node had not behaved well in history. It would not be a good option to select that node as the new VIN. VIN needs to check the next candidate with the 2nd largest TTL value, and so on.

If no hit is obtained in both filters, it implies that a new node is encountered. Here there are two options. The first option is to give up this node and continue to check the status of the node with next highest TTL value. Meanwhile, if we believe that malicious attackers are always the minority among the network population and that a majority of the nodes are willing to cooperate, the second option is to let it be the new VIN. Aside from the largest TTL, the second strategy enables the extension of the cooperator pool. For simulations we employ the second strategy.

In fact, it is challenging when there are only few candidates in the cell, and we get hits only on the D-Filter. What strategy we should take depends on the rules specified by security requirements. For networks that transmit critical sensitive information and cannot tolerate any risk or degrading of service such as violation of confidentiality, data loss or sub-optimal routing, the VIN refuses to transfer the VIN authority and simply takes the steps as though the cell is empty.

IV. Simulation Study

A. Simulation Setup

Network Simulator 2 (NS2) version 2.34 has been adopted to test the performance of LAVI. For the square cell of an area 100 x 100, a 2 x 2 and a 4 x 4 cellular topology are implemented. Although we are testing our protocol with a square topology for simplicity of simulations, in reality a more accurate hexagonal cellular topology can be used like cellular mobile networks.

When all nodes start they are considered NEW nodes. After coming into existence, these nodes transmit Hello packets and wait for Welcome packets. If there is a VIN in the cell it will send the Welcome packet. If NEW nodes receive a Welcome packet they change their status to MEMBER nodes. If a NEW node does not receive a Welcome packet then it assumes that there is no VIN in the cell and declares itself as the VIN of the cell. This process is same when nodes move from one cell to another cell. Each MEMEBR node sends update packets to the VIN to inform it about their current positions as well as the time they will be in that cell.

When a node has data to transmit, it forwards that data to the VIN of that cell. The VIN maintains a register table that contains Node ID and other information such as the location information, time to live value, and status of each member node. If the VIN finds information about a receiver node in its register table, then it forwards the data to that node. If there is no entry for the receiver in the register table then VIN forwards this data to other VINs using multicast addresses.

When a VIN node is moving out of the cell, it hands over its responsibility to a qualified MEMBER. As described in the earlier sections, the VIN maintains a database called C-filter and D-filter to find cooperative nodes and defector nodes. First the VIN checks the TTL value for each member in RT. After finding the node with largest TTL value, the VIN checks whether this node is cooperative. If the node is cooperative, the VIN transfers its RT to this node. After receiving this packet, this node becomes the VIN. In reality, network convergence time will be very small as the new VIN node doesn't have to start building the register table from the scratch.

B. Experimental Results

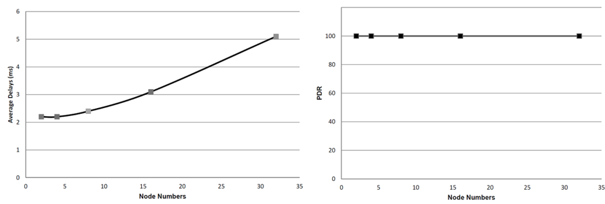
We performed different sets of simulations with various scenarios to test the effectiveness of LAVI. We used mobile nodes available in NS2 to simulate vehicles in VANET. To generate the node movement, we used the setdest utility. These nodes move according to the random waypoint model. We restrict the node movement by disabling random motion and assigning linear movement to nodes. This simulates the motion of vehicles in an urban environment.

The channel type selected was Wireless Channel. The radio propagation model was two ray ground models. Traffic sources were CBR (Constant Bit Rate) with packet size of 128 bytes and transmission interval of 2 sec. For MAC layer we used IEEE 802.11b version available in NS 2. However, in practice, LAVI can be built on other MAC layer standards such as IEEE 802.16 or IEEE 802.11p, which allows a larger cell size. Consequently, it leads to less VIN switching and a more stable network.

We consider Average Delay and Packet Delivery Ratio (PDR) performance metrics to judge the performance of LAVI. The first set of simulations tests the performance of LAVI for different number of nodes in the network. This test verified that LAVI can be effectively scaled to a network with many nodes. The second set of simulations compares the performance of LAVI for 2x2 and 4x4 cellular topologies. This test shows that LAVI can be used as a routing mechanism for mobile ad-hoc networks spanning a very large area with very little degradation in performance. The last set of simulations displays how efficient LAVI is as compared to two popular routing protocols used in MANETs, AODV and DSR.

1) VANET with Different Number of Nodes: We considered 2x2 cellular topology with cell size of 100x100. We ran the simulations with 2, 4, 8, 16 and 32 mobile nodes. All the nodes were given random positions at the beginning of the simulation. All nodes had a speed of 1m/s-5m/s. Each simulation with a different number of nodes was run for 100 sec. CBR source-sink pair starts transmitting 15 sec after simulations begin. Each simulation result we present here is an average of 10 simulations performed with the above settings. Figure 3 shows the average delay and PDR with different node numbers.

As shown in Figure 3(a), the average delay for a scenario with 2 nodes was 0.002 sec for CBR traffic. This is a really good value for applications with constant bit rate. As we increased the number of nodes to 4 and 8, the average delay was still 0.002 sec. For a scenario with 16 nodes, average delay increased a bit to 0.003 sec. Finally, for a scenario with 32 nodes, the average delay rose to 0.005 sec. The increase in average delay is very small and does not bring an obvious impact on the performance of the upper layer application.



(a). Average Delay (ms). (b). Packet Delivery Rate.

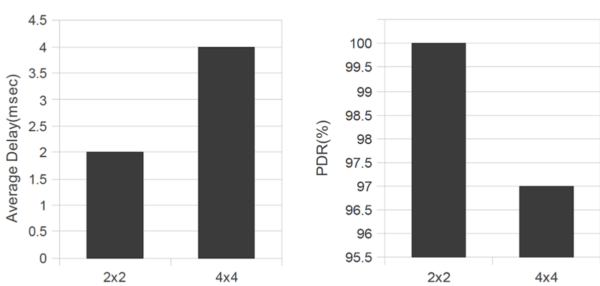
Figure 3. Average Delay and PDR vs. Node Numbers.

We believe that this delay is caused due to more MEMBER nodes trying to communicate with the VIN node, which leads to increased processing time at VIN. This behavior is normal and common with other routing protocols as well. In Figure 3(b), all the CBR packets were delivered to the destination node for all scenarios with a different number of nodes, achieving a PDR of 100%. This PDR might drop when there are more CBR traffic pairs, but this drop will not be much due to the simple but robust routing of LAVI.

*2) **2x2 v/s 4x4 Cellular Topology**:* In this scenario we had two different topologies with a different number of cells. In each case there were 4 mobile nodes. All nodes had varying speeds of 1m/s-5m/s. Random initial locations were assigned to all the nodes.

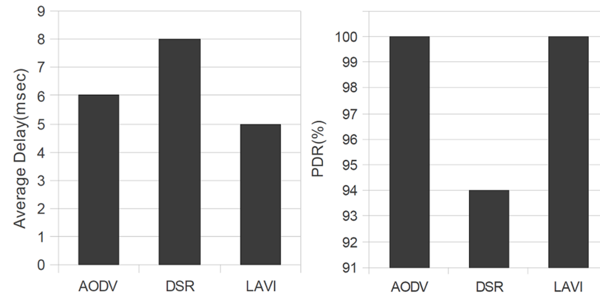
As shown in Figure 4(a), for a 2x2 topology the average delay for CBR traffic was 0.002 sec. When we simulated the scenario with a 4x4 topology the average delay increased to 0.004 sec. However, this small delay was expected and is incurred due to the increased distance in area. Since the source and destination nodes were out of the range, the message was forwarded by the VIN to the VIN of other cells by using multicasting and then to the destination node. Still, this increase was very nominal. This result shows that LAVI can be scaled to larger topologies very easily and efficiently. As shown in Figure 4(b), for a 2x2 topology PDR was again 100%. For a 4x4 topology, one packet was lost during the initial stages of the simulation, thus reducing the PDR to 97%.

*3) **Comparison of LAVI with AODV and DSR**:* In this experiment, we again used the 2x2 square cellular topology with a 100x100 cell size. We used 32 mobile nodes and a CBR traffic pair. All nodes had random initial positions and



(a). Comparison of average delays. (b). Comparison of PDRs.

Figure 4. Average Delay and PDR vs. Cell Numbers.



(a). Average delays of AODV, DSR, and LAVI. (b). PDRs of AODV, DSR, and LAVI.

Figure 5. Comparison of Average Delay and PDR of LAVI with AODV and DSR.

speeds were between 1m/s and 5m/s. Figure 5 presents the experimental results. For a network with 32 nodes and LAVI protocol, the average delay was 0.005 sec, which was the lowest among all three routing protocols considered in this test. For the AODV, the average delay was 0.006. DSR had the lowest value of 0.008 sec.

Figure 5(a) illustrates the difference. It can be seen that LAVI outperformed AODV and DSR in larger networks. Due to the simple routing mechanism of LAVI, it will be more efficient to have routing information even if there are too many nodes. For AODV and DSR, maintaining route information using the RREQ and RREP method for networks with large nodes generates a lot of overhead which can increase delays. AODV and DSR had more delays during the 1st packet forwarding because of the route discovery mechanism with these reactive routing protocols. In terms of PDR, as shown in Figure 5(b), both AODV and LAVI were winners with 100% PDR. DSR lost 2 packets initially reducing its PDR to 94%.

V. DISCUSSIONS

The simulation results verified that our LAVI scheme can compete with current routing protocols in MANET. The average delay during data transmission with LAVI routing protocols is good enough to support applications that are delay sensitive such as video traffic, which is definitely a plus point considering VANET applications. This value is an approximate value, and it can be further reduced.

One of the ways to do that is to implement transmission of the register table to the new VIN during VIN switching. This can reduce the network convergence time and thus new VIN can forward the data quickly. Another solution is to use LAVI with MAC layer technologies that can cover a larger area, such as IEEE 802.16 or IEEE 802.11p standard. According to the radio range of a transceiver, the area can span more than 1 km. With such a large radio range, we can have cells with a large radius which can minimize VIN switching. Hence we can have more stable and robust virtual infrastructure in VANETs that can provide fast service.

As compared to AODV or DSR, LAVI has a very small overhead even when there are more nodes. No matter where the destination is, each source has to contact its local VIN. If the VIN knows about this destination, it forwards the data quickly; otherwise it forwards the data using multicast address which only the VIN of other cells can recognize. In AODV and DSR, major routing overhead for large topologies is due to route discovery processes and route maintaining.

Our cooperation based VIN switching algorithm also provides some level of security rather than selecting new VIN randomly. This can be a plus point for applications requiring security or privacy.

VI. CONCLUSIONS

In this paper, we proposed a novel scheme called location aware virtual infrastructure (LAVI) that constructs a stable logical network infrastructure for VANETs. LAVI screens the complexity resulted from the dynamic network topology. A lightweight but effective evolutionary cooperation mechanism has been adopted that enables a low-cost and transparent VIN switching operation. The major contribution is a stable logical network infrastructure. As illustrated in this paper, in environments where the density of mobile nodes is sufficient, a VANET supported by LAVI can be treated almost as a wireless network without a dynamical topology.

As a future work, we are exploring the ways to reduce the data transmission delay further with LAVI so that it can be used as a robust and effective routing mechanism for delay sensitive applications. We are also looking at simulations of LAVI with larger topologies with more number of mobile nodes and more CBR traffic source-sink pairs. Simulations with TCP traffic is also our interest which can determine effectiveness of LAVI for applications using TCP services. Support for Multicast is also another interesting topic which can be explored further. Specifically, it is of interesting to explore the IEEE 802.11p based LAVI system in the urban environment, where the coverage is closely related to the NLOS (Non-Line-of-Sight) propagation property [6].

REFERENCES

[1] C. A. Aktipis, "Recognition memory and the evolution of cooperation: How simple strategies succeed in an agent-based world," *Adaptive Behavior*, vol. 14, no. 3, pp. 239–247, 2006.

[2] W. Chen, R. K. Guha, T. J. Kwon, J. Lee, and Y.-Y. Hsu, "A survey and challenges in routing and data dissemination in vehicular ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 787–795, 2011.

[3] Y. Chen, C.-J. Wang, W.-S. Ku, and Z. Su, "A location aware virtual infrastructure for vanets," in *Consumer Communications and Networking Conference, Work-in-Progress Workshop (CCNC WiP), 2011 IEEE*. IEEE, 2011, pp. 820–821.

[4] C. Christopher and R. M. Noory, "Wireless channel optimization in vanet by using adaptive modulation," in *ITS Telecommunications (ITST), 2012 12th International Conference on*. IEEE, 2012, pp. 295–299.

[5] K. El Defrawy and G. Tsudik, "Prism: Privacy-friendly routing in suspicious manets (and vanets)," in *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*. IEEE, 2008, pp. 258–267.

[6] J. Gozalvez, M. Sepulcre, and R. Bauza, "Impact of the radio channel modelling on the performance of vanet communication protocols," *Telecommunication Systems*, vol. 50, no. 3, pp. 149–167, 2012.

[7] H. Hartenstein and K. Laberteaux, *vANET: vehicular applications and inter-networking technologies*. Wiley Online Library, 2010, vol. 1.

[8] Y.-L. Hsieh and K. Wang, "Road layout adaptive overlay multicast for urban vehicular ad hoc networks," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*. IEEE, 2011, pp. 1–5.

[9] K. C. Lee, U. Lee, and M. Gerla, "Survey of routing protocols in vehicular ad hoc networks," *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges, IGI Global*, vol. 21, 2009.

[10] S. Manvi, M. Kakkasageri, and C. Mahapurush, "Performance analysis of aodv, dsr, and swarm intelligence routing protocols in vehicular ad hoc network environment," in *Future Computer and Communication, 2009. ICFCC 2009. International Conference on*. IEEE, 2009, pp. 21–25.

[11] A. Patil, Y. Liu, L. Xiao, A.-H. Esfahanian, and L. M. Ni, "Solonet: Sub-optimal location-aided overlay network for manets," in *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*. IEEE, 2004, pp. 324–333.

[12] B. Paul, M. Ibrahim, M. Bikas, and A. Naser, "Experimental analysis of aodv & dsr over tcp & cbr connections with varying speed and node density in vanet," *arXiv preprint arXiv:1204.1206*, 2012.

[13] M. Rezaee and M. Yaghmaee, "Cluster based routing protocol for mobile ad hoc networks," *INFOCOM, Journal of Computer Science*, vol. 8, pp. 30–36, 2009.

[14] G. Samara, W. A. Al-Salihy, and R. Sures, "Security issues and challenges of vehicular ad hoc networks (vanet)," in *New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on*. IEEE, 2010, pp. 393–398.

[15] M. Younis and S. Z. Ozer, "Wireless ad hoc networks: technologies and challenges," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 889–892, 2006.

[16] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (vanets): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[17] G. Zhang, Y. Xu, X. Wang, X. Tian, J. Liu, X. Gan, H. Yu, and L. Qian, "Multicast capacity for vanets with directional antenna and delay constraint," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 4, pp. 818–833, 2012.