ISSN: 2278-0181
NCIECC - 2017 Conference Proceedings

Optimal Routing Technique and Secure Communication in Vanets

Madhu Mathi K B[-1]
Final year
Department of CSE
K.L.N College of Engineering

Rashika V S[-2]
Final year
Department of CSE
K.L.N College of Engineering

Prabhu D[-3]
Assistant Professor
Department of CSE
K.L.N College of Engineering

***Abstract* - Routing is a process of moving a packet of data from source to destination. A Vehicular Ad-hoc Network is a type of Mobile Ad-hoc Network that is used to provide communication between nearby vehicles and between vehicles and fixed infrastructure on the road side. At present the major drawbacks of MANET are overhead which is caused by broadcasting the request message and unstability of routes in the scenario of high mobility. Both these problem can be tackled with abstraction of Virtual Node Layer, which covers the network area with static Virtual Nodes.The intention of this project is to enhance the Virtual Node Ad Hoc On-Demand Distance Vector (VNAODV) to Virtual Cluster Ad Hoc On-Demand Distance Vector (VCAODV) protocol. The packet loss and link breakages can be reduced by this routing method .The security can be provided during packet transmission. The simulation experiments are done using ns3 simulator . It also gives better performance in terms of packet delivery fraction in VANET scenario.The VNLayer and the adaptation of AODV to work with VNs, proved using mathematical analysis . Our solution will achieve better performance in terms of overhead, packet delivery fraction, and latencies in VANET scenarios.**

I. INTRODUCTION

The virtual node layer (VNLayer) was a set of procedures to turn mobile ad hoc networks (MANETs) into more predictable and reliable communication environments. The idea was to engage the mobile *physical nodes* (PNs) in collaboration to emulate *virtual nodes* (VNs) that remain in known grid locations. The geographical area of a MANET is divided into square regions, whose size is chosen so that every PN in a region can send/ receive messages, at least, to/from every other PN in that region and in the neighboring regions. the corresponding region; one of them is chosen as the *leader* in the region and becomes the primary responsible for packet reception, buffering, and forwarding, whereas a subset of *non-leader* nodes is designated as *backups* to maintain replicas of the relevant state information. This way, the VNs can maintain persistent state and be fault tolerant even when individual PNs fail or leave the region as long as there remains at least one PN in the corresponding region.

The key point about the VNLayer is that it resides between the link layer and the Internet layer so that the VNs can be addressed as if they were static server devices. The interface to the network layer exposes the notion of regions, the role played by the corresponding PN at the moment and a set of functions to send/receive messages and to get/set/check state information. This design helps mask the uncertainty that arises from the MANETs' varying topology and from the fact that the physical devices can fail unpredictably. As explained in, the VNs can make it easier for developers to work at the mobile nodes' upper layers since they do not need to deal with the complexities of mobility, location checking, leader election, packet buffering, and state replication. In addition, virtualization creates a level of hierarchy in the otherwise flat MANETs, which brings in opportunities to redesign MANET protocols to operate more efficiently and reliably.

We have been looking at the question of whether the VNLayer constructs could also yield better communications in the more specific (and demanding) realm of VANETs, which are progressively becoming a topic of interest for automotive companies to provide innovative communication services on roads and street. Then, we made experiments using the VANET simulator presented in which revealed several sources of inefficiency, both in the VNLayer constructs and in VNAODV. These are related to features that worked well in MANET scenarios but became inefficient when faced with the mobility models of VANETs, which feature comparatively faster movements conditioned by factors such as road course , urban planning, and traffic regulations. In response to this, we have designed, implemented, and tested a number of refinements that achieve better performance in the vehicular environment. Our contributions to the virtualization layer have to do with the following points:

- A new leader election procedure that involves fewer message exchanges, reduces idle times, and gives priority to the backup nodes that have an up-to-date version of the outgoing leader's state.
- A mechanism to deal with cases of duplicate leaderships, avoiding problems of VN consistency and amplified data traffic.
- A new approach to the designation of backup nodes to enhance the resilience of the VNs and reduce the amount of overhead generated to keep the consistency with the leaders.
- A new mechanism to maintain the state of VNs whose regions run (temporarily) out of supporting PNs.
- Some changes to the interface to the network layer, needed by the aforementioned features and allowing to keep the semantics of link layer services such as the request-to-send (RTS)/clear-to-send (CTS) mechanism of IEEE 802.11.

Our contributions to the network layer, in turn, include the adaptations needed for the virtualized version of AODV to exploit the new features of the virtualization layer and new mechanisms to update the multi-hop routes gracefully in response to the region changes of sources, destinations, and intermediate PNs. The refinements at the virtualization level have been put into a new version of the VNLayer that we called the VaNetLayer, whereas the changes at the networking level lead from the original VNAODV to a new version we will refer to as VCAODV.

The following sections provide a closer look at the procedures of leader election, backup designation, and VN state maintenance. *M_LeaderLeft*: used by a leader node to inform the non-leaders that it is leaving a region. There were also four different timers:

- *T_LeaderRequest*: used to control how long a node waits to send a *LeaderRequest* message after entering a new region or after hearing that the current leader has left;
- *T_RequestWait*: used to control the time a node waits to decide it is not going to hear from a leader after sending an *M_LeaderRequest* message;
- *T_Heartbeat*: used by leaders and non-leaders to control when to send or expect *M_Heartbeat* messages;
- *T_LeaderElect*: used by nonleaders to decide when to start a leader election in the absence of *M_Heartbeat* messages.

In our first approach to using the VNLayer to support communications in VANETs, we replicated the implementations of both the VNLayer and VNAODV as per the extensive details included. To check the validity of our software, we replicated some of the simulation experiments reported for MANETs, and the results were practically identical to those already published. Therefore, we moved on to doing experiments with VANETs using the simulator. Thus, we identified four sources of inefficiency in the VNLayer constructs.

- The leader election procedure reacted very slowly to leader withdrawals (i.e., to the receipt of *M_LeaderLeft* messages), mainly because of the waiting times in the *UNKNOWN* state. This impinged heavily on the communications in scenarios of high mobility since the VNs were down during a non-negligible portion of the average time that the vehicles remained in the respective region.
- The leader election procedure could also designate as new leader either a node that was not acting as a backup or a non-synchronized backup node, even in cases that there were synchronized backups in the region. As a result, the previous exchanges of *M_SyncRequest* and *M_SyncAck* messages became pointless overhead and new synchronizations were triggered immediately. Furthermore, state information from the upper layers was lost unnecessarily, and the routing algorithm on top of the VNLayer could not work properly, inasmuch as the inconsistent data in the routing tables of the different VNs resulted in lost data packets, routing loops, and other problems.
- Due to the probabilistic nature of the backup designation process, there was always a chance that not a single non-leader would choose to work as a backup node. We saw this happen quite often regardless of vehicle density because the probability driving the *CTF* would decrease with the number of nodes in the regions. Having no backup nodes (or too few of them) in a region has a significant impact on the resilience of the VNs. To make matters worse, there were no mechanisms in place to replace outgoing backup nodes by other non-leader nodes in the region.

Only newcomers could take up their place, but the vehicles typical flocking behavior resulted in many situations that the number of backups would decrease even if the number of PNs in the regions increased.

- Finally, we found it critical that the VNLayer does not make any attempts to preserve the state of the VNs when their regions run out of supporting PNs. This may not have been a problem in MANET simulations because the typical mobility models would yield more or less uniform counts of PNs over the different regions. With proper mobility models for VANET studies, in contrast, we found that it is much more likely to have regions depleted temporarily; for example, the presence of vehicles on the extremes of street stretches controlled by traffic lights is very uneven and bursty, i.e., sometimes there are many, and sometimes there are none. This implies an increased probability of VN failures that had a significant impact on the communications.

II. NEW PROCEDURE FOR LEADER ELECTION

Our new procedure for leader election tackles the problem of slow reaction to leader withdrawals by removing the *UNKNOWN* state and rearranging the transitions between the remaining states. It is noticeable that we have also removed the *UNSTABLE* state, but this was just because that state was redundant with the use of a counter of *T_Heartbeat* expirations. The major changes here are as follows.

- First, when non-leaders receive the *M_LeaderLeft* message following the departure of the leader, they change state to *REQUEST* and start their *T_RequestWait* timers. The PN whose timer expires first becomes *LEADER* and immediately sends out an *M_Heartbeat* message to force the other nodes into *NON-LEADER* state. There is no *T_LeaderRequest* as in (which implies less waiting), and the initial values of the *T_RequestWait* are chosen to ensure that the best possible candidate is chosen as a new leader: low values for nodes that were synchronized backups, midway values for non-synchronized backups, and high values for the other non-leaders.

- A new mechanism is in place that allows to completely by pass the *T_RequestWait* timings. This mechanism allows establishing priorities among synchronized backup nodes so that the one with the highest priority (indicated by *PriorityBackup* in can *immediately* transition from *NON-LEADER* to *LEADER* state.

- When a node moves to a new region, its state changes directly to *REQUEST* and immediately applies for leadership by sending out *M_LeaderRequest*. If there was already a leader in the region, the newcomer will receive *M_LeaderReply* and change state to *NON-LEADER*; otherwise, it becomes a leader in much shorter time than possible. This approach speeds up the establishment of VNs in regions that had emptied out and also shortens the downtimes due to lost *M_LeaderLeft* messages: Instead of waiting for two *T_Heartbeat* periods, it suffices to wait until a new node arrives in the region.

III. DEALING WITH DUPLICATE LEADERSHIP

The fact that there are fewer messages and waits involved in leader elections than in the previous version of the VNLayer may entail a greater risk of duplicate leadership in lossy environments. For example, if two nodes, PN X and PN Y, rush into *REQUEST* state at the same time after hearing *M_LeaderLeft* from the previous leader, the one whose *T_RequestWait* timer expires will change to *LEADER* and broadcast *M_Heartbeat*; however, if this message is lost, the *T_RequestWait* timer of PN Y will expire shortly after and PN Y will become *LEADER*, as well. Likewise, a newcomer to a region can get into the *LEADER* state even if there is already a leader in it just because the *M_LeaderReply* that would send it to *NON-LEADER* is lost. With the state machine, duplicate leadership would appear after losses of two messages; with our proposal, only one loss is needed.

Duplicate leadership can be very harmful. For example, if it is a routing algorithm like VNAODV that sits on top of the virtualization layer, duplicate leadership implies that a VN can for-ward the same data message multiple times toward the next hop, causing amplified data traffic. Second, since new self-claimed leaders may not know any routes, data packets sent to them can trigger incorrect data packet drops and unnecessary route discoveries. This disrupts the data forwarding and increases the traffic overhead. Third, when there are multiple leaders in a region that has different upper layer state information, each incoming message could trigger state synchronization in the region, increasing the synchronization overhead, which, in turn, could cause even more duplicate leadership.

IV.BACK TO AODV TRANSMISSION SCHEMES

A major source of problems with VNAODV was the fact that control and data packets could be delivered by broadcast, even if they were aimed at a specific node, i.e., specifically, the VNLayer would use broadcast instead of unicast when the region activeness table of one node did not contain the MAC address of the next hop .

Broadcast made communications more unreliable and made it necessary to implement passive acknowledgments for route maintenance which implies that links can be flagged as broken only after a certain period of idleness.

In consequence, the reporting/repairing actions are started much more slowly than in AODV to the point that they may fail to follow the pace of movement of the vehicles. This problem is aggravated in VANET scenarios since our simulations showed that, due to the characteristic features of node mobility in VANETs, broadcast was used much more often than in MANET settings(a 5%–10% increment).

Plain AODV uses unicast whenever it is known which node is intended to receive one packet, leaving broadcast only for packets that may be processed by any node. Due to the modifications in the interface of the VaNetLayer with regard to the VNLayer, we can follow the same approach in VCAODV, ensuring that control packets such as RREP and RERR and data packets are transmitted via unicast between VN leaders.

To this aim, the entries of the routing tables contain not only the region id of the next VN in the routes (*next_region* field) but also the IP address of the node that is expected to handle the messages there (*next_hop* field).

To preserve the ability of keeping backup nodes and leaders synchronized, we set the link layer in promiscuous mode so that it hands over all the packets it hears in the shared wireless medium to the VaNetLayer, which, in turn, raises them to VCAODV. Up in the network layer, each PN can process the packets as mandated by its role, just like in VNAODV: Leaders and back-ups process the control packets to update their routing tables, leaders forward packets on to the next hops, backups do consistency checks.

This way, VCAODV is less exposed to collisions than VNAODV, it can handle link layer notifications to detect link breakages more quickly, and, finally, it is relieved of the computational burden of keeping track of passive acknowledgments.

New Route Correction Procedure

Our second point of concern about VNAODV relates to the procedure of route correction by destination . This mechanism does serve to avoid a number of link breakages as the destination nodes move, but if we change the transmission schemes of the virtualization layer as previously explained, a subtle flaw appears that causes packet drops in PNs that support heavy traffic. Hop-by-hop forwarding (instead of region-by-region). It is still apparent that route corrections serve to shorten routes and reduce packet delivery times, but there is a shortcoming.

The problem of packet losses with the route correction procedure of Wu *et al.* arises because the entries of the routing tables get new *next_hop* values before the MAC addresses of the corresponding nodes are known. In response to this, we have changed the communications as follows.

ISSN: 2278-0181
NCIECC - 2017 Conference Proceedings

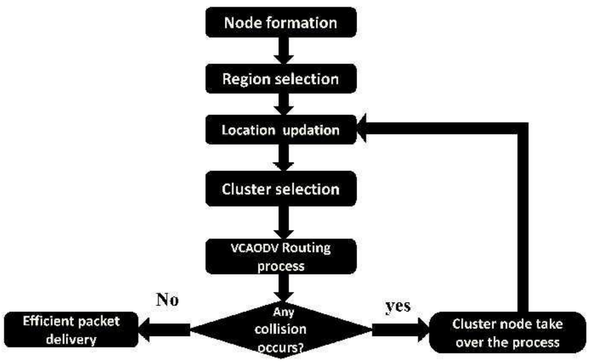
1) When a destination node notices a wrong *next_region* value in a received packet, it broadcasts an RREQ message as if it were trying to discover a route to the source node but indicating TTL = 1.

2) When the RREQ message is processed by the leaders of neighboring VNs, they send RREP messages by unicast to PN B indicating whether and how PN A can be reached through them.

3) Finally, PN B sends one final RREP message via unicast to each one of the replying nodes so that they can update the entry corresponding to PN B in their routing tables.

With this procedure, all of the neighboring leaders will have their routes to the destination node corrected only after they have exchanged RREP messages via unicast with it. Therefore, their region activeness tables will always contain the necessary IP–MAC mappings, and the refreshed routes can be used with no delay. Therefore, we avoid the risk of buffer overflow and packet loss. Admittedly, the exchange of messages takes some time to complete, but this does not cause any trouble because the packets are transmitted through the original routes in the meantime.

Flow Diagram



MODULES

Module 1: (Creation of VANET topology-DSDV Protocol)

Create a Vehicular Network topology with more number of vehicles and RSUs(Road Side Unit) implement IEEE 802.11p protocol and Destination-Sequenced Distance-Vector Routing (DSDV) and use Adhoc On-Demand Distance Vector Routing Protocol to transmit the message from sender vehicle to receiver vehicle.

Module 2: (Implementation of VNAODV)

Create a Vehicular Network topology with more number of vehicles and RSUs(Road Side Unit) implement VNAODV (Virtual Node Adhoc On-Semand Distance Vector) Routing Protocol to transmit the message from sender vehicle to receiver vehicle.

Module 3: (performance analysis of VNAODV)

Packet delivery fraction, Throughput, packet loss are measured for VNAODV and outputs are shown using graphs..

Module 4: (Implementation of Load balancing VNAODV algorithm)

Create a Vehicular Network topology with more number of vehicles and RSUs(Road Side Unit) implement Load balancing VNAODV Routing Protocol to transmit the message from sender vehicle to receiver vehicle.

Module 5: (performance analysis of VCAODV)

Packet delivery fraction, Throughput, packet loss are measured for Load balancing VNAODV Routing Protocol are measured and outputs are shown using graphs.

Module 6: (comparison)
compared VNAODV Routing Protocol with Load balancing VNAODV Routing Protocol using measured parameters (Packet delivery fraction, Throughput,packet loss) and outputs are shown using graphs.

V. SIMULATION EXPERIMENTS AND RESULTS

To validate our proposals, we have made simulation experiment in *ns-3* environment. The *ns-3* (release 3.21) was used to simulate communications based on the IEEE 802.11p standard with wireless signals

In the comparison of the protocol stacks we have looked at the following metrics:

- total overhead, adding up the routing overhead and the virtualization overhead in the cases of VNAODV and VCAODV.

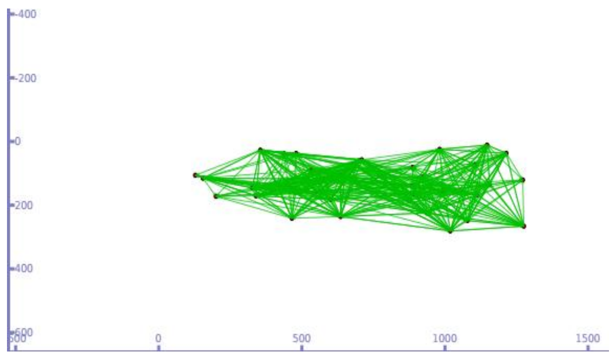


Fig 1: Node communication

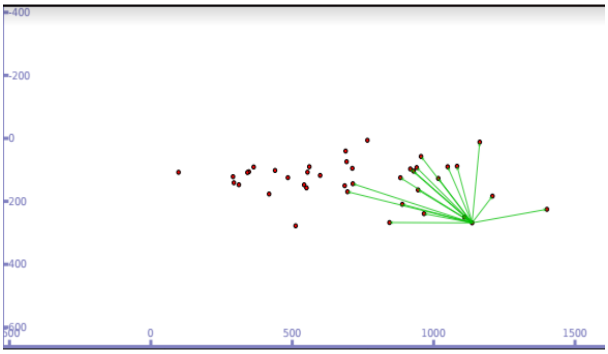


Fig 2: Region selection and route discovery

- packet delivery fraction: the ratio between the number

of packets delivered to the destinations and the number of packets sent by the sources. Packet delivery ratio is directly proportional to the performance.

end-to-end delay: the average time it takes for packets to reach their destinations after they are sent by the sources.

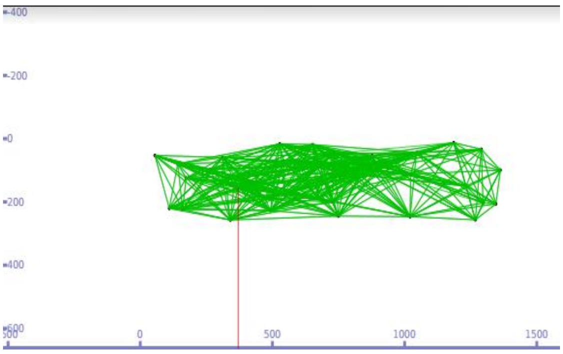


Fig:3 Reduces packet loss during the data transmission to the destination.

VI. CONCLUSION AND FUTURE WORK

we can confirm that the changes we have made to the VNLayer and the VNAODV algorithm do serve to increase the benefits of a virtualization layer when it comes to supporting a reactive routing algorithm like AODV in the demanding realm of VANETs. The new procedures we have put in the VaNetLayer turn the VANETs into more responsive and reliable communication environments than possible with the VNLayer. This comes at the expense of some greater computational cost, i.e., there are several processes running behind the scenes, but this should not be regarded as a problem inasmuch as VANET nodes are not subject to the strict energy, space, and computing capability restrictions of MANETs. In turn, our VCAODV algorithm harnesses the new procedures of the virtualization layer to prevent or resolve route breakages much more effectively than AODV and VNAODV, which ultimately results in greater packet delivery ratios and lower latencies. It is worth noting that, although VCAODV is a refinement of VNAODV, it is not conceptually further from AODV but rather the opposite due to the way it handles unicast and broadcast transmissions and link layer notifications. Overall, our simulation experiments have shown that the combined proposal of having VCAODV on top of the VaNetLayer is competitive with regard to the state-of-the-art in VANET routing, at least in urban scenarios.

REFERENCES

[1] S. Dolev *et al.*, "Virtual mobile nodes for mobile ad hoc networks," in *Proc. 18th Annu. DISC*, Amsterdam, The Netherlands, Oct. 2004, pp. 230–244.

[2] M. Brown *et al.*, "The virtual node layer: A programming abstraction for wireless sensor networks," *ACM SIGBED Rev.*, vol. 4, no. 3, pp. 121–140, Jul. 2007.

[3] J. Wu, N. Griffeth, C. Newport, and N. Lynch, "Engineering the virtual node layer for reactive MANET routing," in *Proc. 10th IEEE Int. Symp. NCA*, Cambridge, MA, USA, Aug. 2011, pp. 131–138.

[4] R. Patil and S. R. Shah, "Cross layer based virtual node layer for reactive MANET routing," *Int. J. Eng. Res. Technol.*, vol. 1, no. 6, pp. 1–7, Aug. 2012.

[5] C. E. Perkins, E. M. Royer, and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) routing," 2003. [Online]. Available: http://www.ietf.org/ rfc/rfc3561.txt

[6] M. Gerla and L. Kleinrock, "Vehicular networks and the future of the mobile Internet," *Comput. Netw.*, vol. 55, no. 2, pp. 457–469, Feb. 2011.

[7] J. Wu, "A simulation study on using the virtual node layer to implement efficient and reliable MANET protocols," Ph.D. dissertation, Graduate Faculty Comput. Sci., City Univ. New York, New York, NY, USA, 2011. [Online]. Available: http://groups.csail.mit.edu/tds/papers/Wu/ JiangWuThesisFinal.pdf

[8] J. Bravo-Torres, M. López-Nores, Y. Blanco-Fernández, and J. Pazos-Arias, "On the use of virtual mobile nodes with real-world considerations in vehicular ad hoc networks," in *Proc. 9th Int. COMM*, Bucharest, Romania, Jun. 2012, pp. 193–196.

[9] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks: A survey and taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 4, pp. 19–41, 4th Quart. 2009.

[10] L. Kleinrock, *Queueing Systems*, vol. I, *Theory*. Hoboken, NJ, USA: Wiley Interscience, 1975.

[11] G. S. Malkin, *RIP: An Intra-Domain Routing Protocol*. Reading, MA, USA: Addison-Wesley, 2000.

[12] R. Droms, "Dynamic host configuration protocol," 1997. [Online]. Avail-able: http://www.ietf.org/rfc/rfc2131.txt

[13] N. Lynch, S. Mitra, and T. Nolte, "Motion coordination using virtual nodes," in *Proc. 44th IEEE CDC-ECC*, 2005, pp. 2823–2828.

[14] M. Saleem, E. Renault, and D. Zeghlache, "Information centric network-ing based handover support for QoS maintenance in cooperative heteroge-neous wireless networks," 2011. [Online]. Available: http://arxiv.org/pdf/ 1108.3239v1.pdf

[15] W. Zhao and M. H. Amma, "Message ferrying: Proactive routing in highly partitioned wireless ad hoc networks," in *Proc. 9th IEEE Workshop Future Trends Distrib. Comput. Syst.*, San Juan, Puerto Rico, May 2003, pp. 308–314.

[16] Q. Binbin and W. Jie, "Multi-path intelligent virtual mobile nodes for ad hoc networks," in *Proc. 17th ICCCN*, St. Thomas, U.S. Virgin Islands, Aug. 2008, pp. 1–8.

[17] M. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," *Wireless Netw.*, vol. 1, no. 1, pp. 61–81, Feb. 1995.

[18] A. Lee, I. Ra, and H. Kim, "Performance study of ad hoc routing protocols with gossip-based approach," in *Proc. SpringSim*, San Diego, CA, USA, Mar. 2009, p. 89.
[19] O. Kayis and T. Acarman, "Clustering formation for inter-vehicle com-munication," in *Proc. 10th IEEE ITSC*, Seattle, WA, USA, Oct. 2007,
pp. 636–641.
[20] C. Shea, B. Hassanabadi, and S. Valaee, "Mobility-based clustering in VANETs using affinity propagation," in *Proc. IEEE GLOBECOM*, Honolulu, HI, USA, Dec. 2009, pp. 1–6.
[21] M. Almalag and M. Weigle, "Using traffic flow for cluster formation in vehicular ad-hoc networks," in *Proc. IEEE 35th Conf. LCN*, Denver, CO, USA, Oct. 2010, pp. 631–636.
[22] S.-S. Wang and Y.-S. Lin, "PassCAR: A passive clustering aided routing protocol for vehicular ad hoc networks," *Comput. Commun.*, vol. 36, no. 2, pp. 170–179, Jan. 2013.
[23] Network Simulator (ns-3). [Online]. Available: http://www.nsnam.org/
[24] K. Mahler, P. Paschalidis, A. Kortke, M. Peter, and W. Keusgen, "Realistic IEEE 802.11p transmission simulations based on channel sounder measurement data," in *Proc. 78th IEEE VTC—Fall*, Sep. 2013,
pp. 1–5.
[25] J. Harri, F. Filali, and C. Bonnet, "Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns," in *Proc. 5th IFIP Med-Hoc-Net Workshop*, Lipari, Italy, Jun. 2006, pp. 1–8.
[26] K. Lee, U. Lee, and M. Gerla, "Survey of routing protocols in vehic-ular ad hoc networks," in *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges*. Hershey, PA, USA: IGI Global