Implementing Clustering for Vehicular Ad-hoc Networks in ns-3

Conference Paper · May 2015

DOI: 10.1145/2756509.2756520

3 authors, including:

Lampros Katsikas
National and Kapodistrian University of Athens

Konstantinos Chatzikokolakis
MarineTraffic

Implementing Clustering for Vehicular Ad-hoc Networks in ns-3

Lampros Katsikas
National and Kapodistrian University of Athens
lkatsikas@di.uoa.gr

Konstantinos Chatzikokolakis
National and Kapodistian University of Athens
kchatzi@di.uoa.gr

Nancy Alonistioti
National And Kapodistrian University of Athens
nancy@di.uoa.gr

ABSTRACT
Vehicular Ad-Hoc Networks are of great interest for secure, efficient and reliable communications, in the last years. In this paper, we firstly present an implementation of a VANET algorithm to form clusters in ns-3 simulation environment. We provide the details of the algorithm and the communication messages exchanged among vehicles to form and maintain clusters of moving nodes. Then, we present an application scenario for safety and emergency situations. Vehicles form clusters using the proposed algorithm while the application calculates safety messages statistics.

Categories and Subject Descriptors
C.2.1 [**Network Architecture and Design**]: Network communications; I.6 [**Simulation and Modeling**]: General, Applications, Model Development,

General Terms
Algorithms, Experimentation, Measurement

Keywords
ns-3, Simulation, Clustering Schemes, V2V, Safety, IEEE 802.11p

1. INTRODUCTION
There is a big trend in vehicles communication over the last years and especially in vehicle applications. Vehicular Ad-hoc NETworks (VANET) consist mainly of two types of communication, vehicle-to-vehicle (V2V), and vehicle-to-infrastructure (V2I) [1]. V2V communication networks are formed by moving vehicles, equipped with wireless interfaces, which allow vehicles to exchange communication messages. This type of communication is most suited for short-range communication networks. V2I networks make use of preexisting network infrastructure such as Road-Side Units (RSUs). Communication between vehicles and RSUs is supported by protocol best suited for longer communication range.

VANET applications could also be divided into i) Safety applications (e.g. Incident notification, road hazard notification) in which messages should be exchanged quickly and reliably and thus such applications could be characterized as delay critical, ii) Traffic management applications (e.g. collision avoidance, root trip management, etc.) which aim to reduce time and fuel consumption of the vehicles by changing routes, and iii) Infotainment applications (e.g. video games, emails) the so called value added services [2]. Cooperative Collision Warning (CCW) could be considered as a sub-domain of safety applications. The main idea behind this type of applications is to prevent collisions and accidents by broadcasting warning messages to the neighboring vehicles in time. CCW is characterized also by periodic messages broadcasting information such as location, velocity, control information, so that neighbors can get a snapshot of their current status [3].

A V2V communication commonly takes place, using 7 channels of 75 MHz in total, in the 5.9 GHz band which is licensed for the Dedicated Short Range Communication (DSRC) [4]. Latest version of the ns-3 simulator [5] maintains an implementation of the IEEE 802.11p standard, widely known as Wireless Access in Vehicular Environments (WAVE) [6]. However, a large amount of devices may attempt to cooperate in this band and could lead to communication failure due to congestion when using the aforementioned protocol. Clustering schemes might be a solution to the network congestion problem by reducing data volumes exchanged. However, the absence of a clustering mechanism in the simulator is the motivation for our work. Our implementation extends version 3.21 of the ns-3 simulator, providing a new application model that implements a V2V clustering algorithm [7]. Apart from the algorithm implementation, we also provide a CCW safety application example that exploits the clustering scheme to group vehicles and disseminate the events to the moving nodes. The source code of the implementation is publicly available as well as the example code so as to reproduce the configuration and the results that are described in this paper[1].

The rest of this paper is organized as follows: Section 2 provides an overview of the most important V2V clustering algorithms that could be found in the literature. Section 3 describes the implementation details of the current work. Section 4 evaluates the performance of the model and analyzes the results of simulation. Section 5 concludes this paper.

WNS3 2015, May 13 2015, Barcelona, Spain
© 2015 ACM. ISBN 978-1-4503-3375-7/15/05...$15.00 DOI: http://dx.doi.org/10.1145/2756509.2756520

[1] Available at https://gitlab.scanlab.gr/lkatsikas/v2v.git

2. RELATED WORK

In this section we provide the state of the art approaches of V2V Clustering Schemes that appear in the literature. In [8] a novel approach using Affinity Propagation algorithm as clustering criterion that minimizes both relative mobility and distance between Cluster-Head (CH) and Cluster-Members (CMs) is introduced. The information exchange among neighboring nodes involves packets regarding two parameters: i) responsibility and ii) availability. Responsibility represents node's suitability to become CH and availability indicates the desire of a node to become a CH. When Global Positioning Systems (GPS) measurements are available, stability is further increased based on moving direction of vehicles based on a similarity function that evaluates current and future vehicles' positions. In [9] the authors propose an algorithm designed for aeronautical ad hoc networks, which, however, can also be applied in highway scenarios. Direction and velocity of nodes are being considered as inputs for the clustering algorithm. Their criterion for cluster formation is based on the time period nodes are expected to lie within communication range. On the one hand, when nodes' location is not available, they rely on the Doppler shift from control packets exchanged among nodes so as to calculate this time period. On the other hand, when nodes can obtain their current location, this time period is calculated based on Link Expiration Time (LET) [10]. Clusters consist of nodes that lie within communication range for at least a minimum threshold. The proposed solution in [11] focuses on long cluster lifetime where the participating nodes calculate their average velocity and acceleration and through message broadcasting all nodes receive the corresponding values of their neighbors. Based on the exchanged information, each node is able to calculate its Spatial Dependency [12] with each neighbor, its total Spatial Dependency regarding its entire neighborhood and its Cluster Relation with its neighbors. The last metric is used as criterion for the cluster formation. In [13] the vehicles are equipped with digital maps split into smaller regions and are able to obtain their direction which is then used as input in cluster formation. Vehicles with the same direction are grouped in the same cluster; however, clusters are recalculated at intersections due to vehicles moving towards different directions while exiting an intersection. Packets/messages are successfully exchanged only among vehicles travelling to the same direction (i.e., vehicles within the same cluster). The Clustering Algorithm for VANETs introduced in [14] is based on Basagni's Distributed and Mobility-Adaptive Clustering (DMAC) algorithm presented in [15]. In this VANET-oriented version, each vehicle calculates a weighted value indicating its suitability to become CH (higher value represents higher suitability). This value is calculated based on parameters such as position, velocity, and connectivity. In general, vehicles that become CHs, retain this value high as clusters move and consequently re-clustering is avoided and clusters' lifetime is extended. In [16], the communication among vehicles is unreliable in sparse networks and thus the algorithm's behavior is affected. There are algorithms [17] that also consider the behavior of vehicles during message forwarding but such metric increases significantly the complexity of the system. In [18] overlapping clusters are created but this approach leads to unnecessary traffic exchanged. In [19], the authors propose a mechanism that focuses on fast cluster formation rather than focusing on clusters' stability. In [20] the authors assume that the number of lanes of a road and the lane each vehicle resides is known, which however, is not always the case in real environments.

3. IMPLEMENTATION

In this section, we firstly give a detailed description of the algorithm that we have selected for our implementation, providing the basic steps and the rules for the cluster formation and for the cluster maintenance respectively. Then, an implementation of a UDP application model namely V2vControlApplication, that extends the built-in Application model of ns-3 simulator is presented. Figure 1 shows the dependency graph of the V2vControlApplication class. In addition, we present the implementation of the new header messages that assist vehicles' communication so as to form and maintain the clusters efficiently.

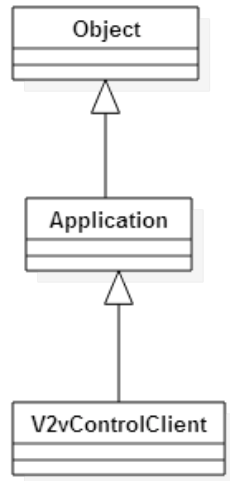

Figure 1: The inheritance diagram of the V2vControlClient application model.

Figure 2 shows the dependency graph of the new communication header messages. V2vIncidentEventHeader contains the information sent when an incident occurs, V2vClusterInfoHeader is used for the information update of the neighbors while V2vFormClusterHeader and V2vInitiateClusterHeader are messages that are exchanged during the cluster formation steps. The last part of this section consists of a case study example that exploits the clustering algorithm to form clusters and assess its performance against key performance indicators for emergency situations, such as the delay of the messages exchanged when an incident occurs.

3.1 Algorithm Description

We have selected the algorithm in [7] for our implementation, which is a well-established algorithm for forming clusters and enhancing the stability of the network topology. The degree of the speed deviation among neighboring vehicles is the key criterion in order to construct stable clusters. Neighboring vehicles are defined as stable or unstable clustering neighbors based on their velocity vector (i.e. speed and direction) and only neighbors that are stable may form clusters. In addition, any two vehicles could be divided into r-Stable and 2r-Stable according to their r distance.

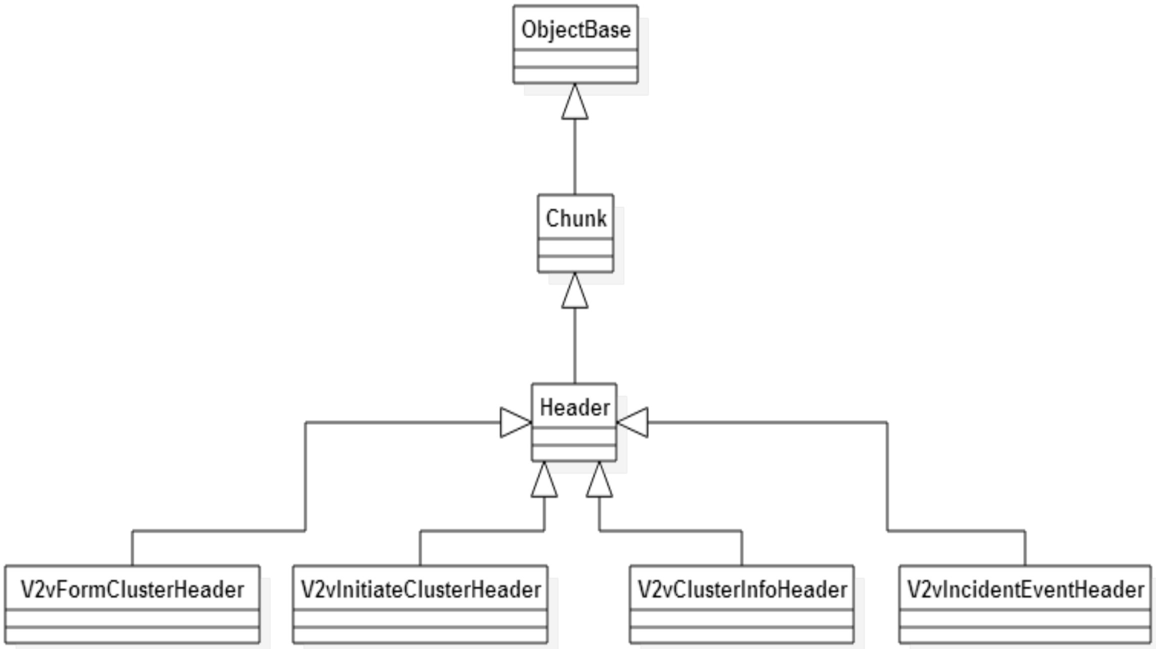

Figure 2: Inheritance diagram of the new header models.

3.1.1 Cluster Formation

In more detail, vehicles broadcast their current mobility state using periodic messages. The content of the different kind of messages is presented with details in subsection 3.3. Neighbors with relative speed less than a predefined threshold could be considered as stable neighbors. The cluster formation process is initiated when the slowest vehicle sends a cluster-originating message. All 2r-Stable neighbors with greater speed react on this message reception by changing their cluster identifier (ID) temporarily to the ID of the originating node and by calculating their suitability value to become CH. The suitability value of each vehicle is proportional to the deviation of its position and velocity compared to the mean values calculated by the vehicle's stable neighbors. This value reflects the time that the vehicle has to wait before announcing its eligibility to become cluster-head by sending a message to form the cluster. Vehicles that receive such message before their waiting time expires, quit the cluster-head election competition, set their state to CM and change their temporary cluster ID to the new cluster ID embedded in the received message. The rest of the neighboring vehicles that were not part of this process will go through the same cluster formation process to create other clusters.

3.1.2 Cluster Maintenance

Due to high mobility, communication messages in VANETs could lead to excessive signaling. It is of great importance to minimize the number of cluster changes by minimizing vehicle transitions among clusters and thus, the exchanged messages. Thus, the stability of the cluster is considered as a key parameter for the algorithm implementation. In this section we analyze the basic rules for the cluster maintenance according to the following events:

- Joining a cluster: This event could be divided into two sub-cases. If there is only one CH in the vicinity area then a potential CH will accept the vehicle in case their speed deviation is lower than a predefined threshold. In case of more than one cluster-heads, the vehicle joins the cluster that it will remain for the longest time period. This is also called greatest remaining time (RT). This metric is calculated based on speed, direction and position information that nodes already have for their neighbors.
- Leaving a cluster: If there is neither cluster-head in the vicinity area to attach, nor standalone vehicles to form new cluster, then the vehicle changes its state to standalone and leaves the cluster.
- Cluster merging: In case that two CHs come within each other's range and their speed deviation is within a predefined threshold then a cluster merge event may be triggered. More specifically, the CH with less number of members gives up its CH role and becomes CM in the other cluster. The other members also join if they are within the new cluster-head's range, or they calculate the greatest RT before joining a cluster in case of more than one cluster heads existing in the vicinity area. Otherwise, if there is no cluster in range for some standalone vehicles, then those vehicles start a clustering process to form new cluster.

3.2 V2vControlApplication Model

The full implementation of the algorithm can be found in v2v-control-client.cc class file while the prototype of functions and variables are located in the correspondent header v2v-control-client.h file. This class extends the Application built-in class of ns-3 simulator, which means that virtual methods of the class have been overridden to provide the new functionality. The following sections analyze the functionality that provides the class described above.

3.2.1 Application Methods

Every vehicle, according to the algorithm, should be a transceiver and thus it should be able both to send and receive messages. For this purpose one socket for transmitting packets and one socket listening for new packet arrivals have been created and are handled from Send and HandleRead methods respectively. The key role of the Send method is to create the appropriate type of header messages in each transmission and fill the contents of the header, while the HandleRead method is responsible to read and process the different type of header messages accordingly. The implementation of an access protocol is out of scope of this work. Thus, we have modeled a TDMA scheme for this purpose and the time window that each node sends its update message is given as a parameter to our model.

3.2.2 Vehicle States

Cluster formation could be characterized as sequential procedure since a couple of steps should take place in order to finalize the grouping of the vehicles. Consequently, we have modeled this process as a Finite State Machine (FSM) with four different states:

- CLUSTER_INITIALIZATION: The vehicle is in a STANDALONE state out of any cluster. A periodic message broadcasting vehicle's mobility and status information is the only communication message that is permitted in this state.
- CLUSTER_HEAD_ELECTION: After the flooding of the initial information messages, vehicles have obtained mobility information for their neighbors and thus, the slowest vehicle starts the cluster formation process by sending an Initiate Cluster message. The receivers of this message start a process to calculate their capability to become CH.
- CLUSTER_FORMATION: The most capable vehicle to become CH (i.e. the vehicle with the lowest suitability value) sends a Form Cluster Message as the elected CH and the receivers of this message set themselves as CM of the cluster.
- CLUSTER_UPDATE: After the cluster formation, every node schedules a periodic event to send update information to its neighbors about its new mobility information.

3.2.3 Neighbor Information

An important part of the mechanism is the maintenance of the information received from the neighbors. Each vehicle maintains a map structure with the information received from the periodic update messages. This structure consists of mobility information (speed, position, direction) the degree of the vehicle (STANDALONE/CM/CH), the ID of the vehicle and the ID of the cluster that is part of. In order to maintain a valid and updated set of information of the neighboring nodes, vehicles schedule an internal update information event after the update process. The aim of this event is to remove the nodes that are not in range so far, trigger a possible cluster remove and search if there are other cluster-heads in range.

3.2.4 Algorithm Specific Processes

This section provides a description of specific processes that implement basic functionality of the algorithm used.

- IsStable: This method is called whenever there is a new update message arrival. The method returns true if the neighbor could be marked as stable according to its relative position and speed.
- IsSlowest: The slowest vehicle in range can start the formation process sending an Initiate Cluster message. This method is triggered exclusively in the cluster formation process.
- MergeCheck: The role of this method is to find the CH that has the best RT, which is calculated from the mobility information of the vehicle and its neighbors as we described in Section 3.1.2.
- SuitabilityCheck: This method is responsible to calculate the suitability of a vehicle to become CH. The calculation is based on the position and the speed of the vehicle compared to the mean values of position and speed of its stable neighbors. This method is used in cluster-head election process.
- UpdateNeighborList: This method is called periodically after the update process setting the current neighbors of the vehicle. This method might also trigger a cluster-head change or a leave cluster event.

3.3 V2vHeader Model

The implementation of the new header messages that we have developed is located in the v2v-cluster-header.cc class file. The corresponding v2v-cluster-header.h file contains the functions and variable prototypes as usual for ns-3 extensions. We have developed the three following types of header messages to support the cluster formation process.

- V2vClusterInfoHeader: This type of message contains a structure with the mobility information of the node (i.e., position, speed, direction), the degree of the node, vehicle's ID and the cluster ID that the vehicle is attached to. This type of header message is also used in the cluster update process.
- V2vInitiateClusterHeader: This message contains only the temporary cluster ID, since it is the first message that starts the cluster formation process. It is originated from the slowest vehicle in range.
- V2vFormClusterHeader: This message contains information corresponding to V2vClusterInfoHeader described above. This type of message is broadcasted from the newly elected cluster-head to the members of the cluster.

It is important to highlight that all the header messages above also include a Timestamp field, which is assigned in the creation of the packet each time.

3.4 Case Study: Safety Application

The v2v-clustering-example.cc file is the implementation of our test case scenario. The main concept of this class is to use the implemented clustering scheme in the scope of a safety application. The following two sections describe the details of our safety application scenario and the extensions required to the clustering model respectively.

3.4.1 Scenario Description

The safety application scenario aims to demonstrate the consistency of the clustering mechanism and assess its performance against key performance indicators for emergency situations, such as the delay of the messages exchanged when an incident occurs. In more details, the scenario consists of a number of vehicles positioned in a 3-lane road with 3 meters inter-lane distance. The vehicles start moving towards one direction. The velocity that is assigned to each vehicle is a uniform random variable with a mean value of 30 m/s and 5 m/s deviation. The clustering algorithm is applied in the first stage of the simulation. After the information exchange is complete the vehicles are grouped into clusters. The periodic exchange of mobility information assures that the cluster structure is kept updated. The cluster maintenance rules are triggered according to the algorithm description in Section 3.1.2.

Each vehicle additionally supports the transmission of incident events. This new feature is an add-on to the clustering algorithm implemented by a new type of header message. Those messages are triggered randomly for each vehicle during the simulation. When a vehicle identifies the occurrence of such incident (this could be captured by e.g. sudden break or based on sensor data monitored by the vehicle), it opens a point-to-point connection socket with the cluster-head so as to send a message containing this event description. Then, the cluster-head broadcasts the received message to the rest of the CMs. In our implementation each node that sends an incident event, keeps track of the time that this event occurred until the time that the broadcasted message from the CH has been received, in order to measure the delay the nodes experience. It is self-explanatory that the model supports also the destruction of the point-to-point socket between a vehicle and its CH for the incident event forwarding as well the creation of a new one when a cluster head change event takes place.

3.4.2 Incident Description

The new header message that we have developed for an incident report purpose is simulated with V2vIncidentEventHeader class. This class implementation is similar to the implementation of the other header messages that have been described in section 3.3 in terms of extending the Header class of ns-3. The information that is packed within a V2vIncidentEventHeader message consists of the timestamp and the type of the event. In the current implementation the type of the event could be either a NOTIFICATION_EVENT or an EMERGENCY_EVENT. Contrariwise to the other header messages, this type of message is not transmitted in broadcast mode. By the time an incident event occurs, the vehicle could be in 3 possible states, namely STANDALONE, CM or CH. If the vehicle is in CH or STANDALONE mode, then it broadcasts the event. If the vehicle is in CM mode sends the message directly to the CH, which then, is responsible to broadcast this message to the whole cluster group once received.

4. EVALUATION

In this section we provide the evaluation of the clustering mechanism and assess its performance against key performance indicators related to emergency situations. As stated before, such scenarios demand very reliable communication and low end to end delay, so that the following vehicles can effectively perceive the situation and react on time. In order to evaluate the clustering algorithm we have studied how variations in the WAVE parameters (e.g. Tx Gain, Rx Gain, Tx Power) affect both the cluster formation process and the stability of the clusters. Then, we have measured the delay of the incident messages broadcasted only from the CMs of the clusters. This is the worst case scenario in terms of delay measurement because this case consists of the delay of the message to reach the CH and the delay to be broadcasted from the CH to the CMs. One important parameter for VANET simulation scenarios is the mobility model of the vehicles. However, the pre-existing mobility models in the ns-3 simulator could not fit to our scenario, since there is a need of a non-constant velocity and constant direction model for vehicles, which provides information for position, speed and direction. For this purpose we provide an implementation of a new mobility model that simulates the movement of the vehicles (i.e. check v2v-mobility-model.cc file). Table 1 illustrates the configuration of the basic parameters of the scenarios.

Table 1: Basic configuration parameters of the clustering scenarios

Parameter	Value
Mean Velocity	30 m/s
Velocity Deviation (per vehicle)	5 m/s
Lane Distance	3 m
Number of Vehicles	10/25/50
Simulation Time	30 s

4.1 Clustering Process Scenario

The scope of those simulation runs is to indicate how the clustering process is affected by the communication parameters of the WAVE protocol such as the transmission power, the transmission and the reception gain which directly affect the radio range of the vehicles. The messages exchanged among vehicles and the calculation of the frequency that vehicles change, create or leave clusters, are two basic metrics for the evaluation of the clustering algorithm. Table 2 contains the WAVE parameters that we have configured in each scenario.

Table 2: WAVE configuration parameters of the clustering scenarios

Parameter	Scenario 1	Scenario 2	Scenario 3
Tx Power	24 dBm	28 dBm	32 dBm
Tx Gain	6 dB	9 dBm	12 dBm
Rx Gain	6 dB	9 dBm	12 dBm

According to Table 2 there are three different scenarios, which differ in the transmission range due to the different value of the Tx Power and Tx/Rx Gains that have been selected. Each scenario has been tested with three different numbers of vehicles, 10, 25 and 50 vehicles respectively. The simulations reveal that the performance of the clustering algorithm both in formation and maintenance scope is proportional to the transmission range of the vehicles.

4.1.1 Cluster Formation

Formation messages consist of three types of messages. One message for the mobility awareness of the neighboring vehicles, one message that is sent only from the slowest node in the range to start the formation process, and one last message that is sent from the elected CH. Figure 3 shows the results of the simulation of the cluster formation process. We observe that increasing the transmission range of the vehicles, we can slightly reduce the number of messages exchanged in the formation process.. Considering that the messages are broadcasted, the gain of this reduction could be higher. It is important also to note that for a small number of vehicles the reduction is imperceptible.

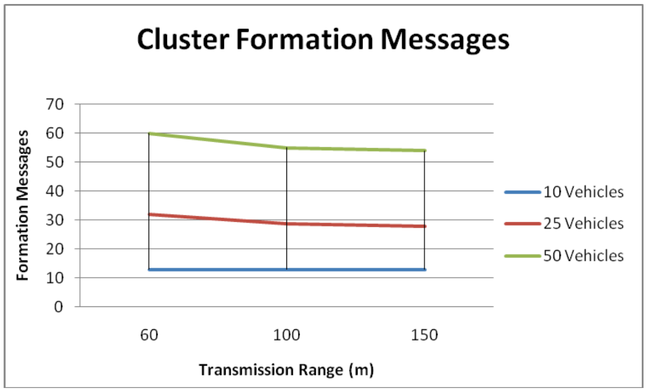

Figure 3: Cluster Formation Scenario

4.1.2 Cluster Stability

Cluster stability is a key indicator for a clustering implementation. Since there is no other algorithm implementation to be compared, our evaluation consists of the calculation of the frequency that each vehicle changed its state during the simulation. In other words, we calculate the times that the vehicles join, leave or merge with other vehicles tested in different transmission range scenarios and with varying number of vehicles. Figure 4 shows the results of the scenario described above. The results demonstrate that the stability of the cluster increases when the range of the cluster increases. This is the expected behavior since a vehicle with greater transmission range can exchange messages with more neighbors, and thus the probability to find a stable neighbor in the vicinity area is greater.

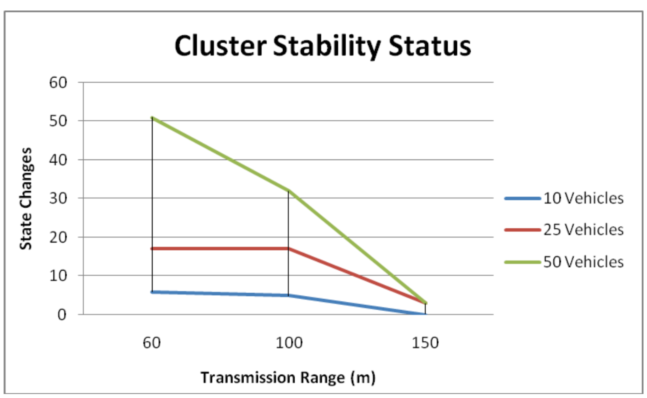

Figure 4: Cluster Stability Scenario

4.2 Safety Messages Delay Scenario

Using the same configuration parameters with the previous clustering scenarios, in this scenario we measure the delay of the incident reports. As mentioned above, we have evaluated the worst-case scenario, which occurs when a CM identifies an incident, it reports it to the CH, which then, broadcasts the message to the whole cluster. Figure 5 shows the average delay measured according to the transmission range and the number of vehicles tested. The results indicate that the number and the transmission range of the vehicles play a key role in the delay. Increasing the number of vehicles also increases the delay but we observe that increasing the transmission range can reduce the messages delay. Increasing the transmission range of the cluster improves vehicles' grouping and therefore vehicles maintain a better relative position to their CH leading thus, to decreased delay.

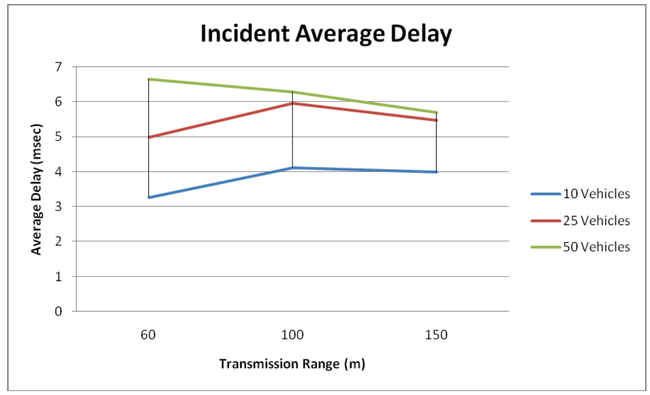

Figure 5: Incident Messages Scenario

5. CONCLUSION

It is expected that future cellular and wireless networks will comprise a variety of communication scenarios with diverse requirements. V2V communication is such a paradigm that has emerged recently. The lack of V2V communication in ns-3 simulator till version 3.21 gave us the motive to start the implementation of a V2V application but also for an implementation of a clustering algorithm for VANETs. In this paper, we have provided an implementation of an algorithm suitable to form stable clusters [7]. Since our implementation could not be compared directly with another approach, we provide indicative results both for the cluster formation and for the stability of the cluster algorithm. Moreover, we extended our implementation to support a CCW application scenario where the vehicles provide results for the delay of the incident messages transmitted. In future we are planning to extend this application by replacing the current mobility model with a more realistic mobility pattern using SUMO [21] traffic simulator. Moreover we are keen on providing further results for the current clustering algorithm, for the delay of the incident reports and perform more extensive testing with greater number of vehicles.

6. REFERENCES

[1] S. Zeadally, R. Hunt and Y. S. Chen. Vehicular ad hoc networks (VANETS): status, results, and challenges. In Telecommunication Systems, August 2012, 50(4), 217-241

[2] H. Hartenstein and K. P. Laberteaux. A Tutorial Survey on Vehicular Ad Hoc Networks. IEEE Communications Magazine, June 2008, 46(6), 164-171

[3] T. ElBatt, S. K. Goel, G. Holland, H. Krishnan and J. Parikh. Cooperative collision warning using dedicated short range wireless communications. Proceedings of the 3rd international workshop on vehicular ad hoc networks, September 2006, 1-9

[4] J. B. Kenney. Dedicated Short-Range Communications (DSRC) Standards in the united states. Proceedings of the IEEE, July 2011, 99(7), 1162-1182

[5] Network Simulator-3 (NS-3)

[6] J. Bu, G. Tan, N. Ding and M. Liu. Implementation and Evaluation of WAVE 1909.4/802.11p in ns-3. Proceedings of the 2014 Workshop on ns-3, Atlanta, Georgia, USA, May 2014.

[7] Z. Y. Rawashdeh. A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. EURASIP Journal on Wireless Communications and Networking, January 2012

[8] B. Hassanabadi, C. Shea, L. Zhang and S. Valaee. Clustering in Vehicular Ad Hoc Networks using Affinity Propagation. Ad Hoc Networks, February 2014, 13(B), 535-548

[9] E. Sakhaee and A. Jamalipour. Stable Clustering and Communications in Pseudolinear Highly Mobile Ad Hoc Networks. Vehicular Technology, IEEE Transactions, November 2008, 57(6), 3769-3777

[10] W. Su, S. J. Lee and M. Gerla. Mobility prediction in wireless networks. MILCOM 2000, 21st Century Military Communications Conference Proceedings, 1, 491-495

[11] W. Fan, Y. Shi, S. Chen and L. Zou. A mobility metrics based dynamic clustering algorithm for VANETs. Communication Technology and Application (ICCTA 2011), IET International Conference, October 2011, 752- 756

[12] F. Bai, N. Sadagopan and A. Helmy. IMPORTANT: a framework to systematically analyze the Impact of Mobility on Performance of Routing Protocols for Adhoc Networks. INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, April 2003, 2, 825-835

[13] N. Maslekar, M. Boussedjra, J. Mouzna and L. Houda. Direction based clustering algorithm for data dissemination in vehicular networks. Vehicular Networking Conference (VNC), October 2009, 1-6

[14] G. Wolny. Modified DMAC Clustering Algorithm for VANETs. Systems and Networks Communications. ICSNC '08. 3rd International Conference, October 2008, 268-273

[15] S. Basagni. Distributed clustering for ad hoc networks. Parallel Architectures, Algorithms, and Networks. (I-SPAN '99) Proceedings. Fourth International Symposium, June 1999, 310-315

[16] S. Kukliński and G. Wolny. Density based clustering algorithm for VANETs. Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops. TridentCom 2009. 5th International Conference, April 2009, 1-6

[17] A. Daeinabi, A. G. P. Rahbar and A. Khademzadeh. VWCA: An efficient clustering algorithm in vehicular ad hoc networks. Journal of Network and Computer Applications, January 2011, 34(1), 207-222

[18] R. T. Goonewardene, F.H. Ali and E. Stipidis. Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks. Intelligent Transport Systems, June 2009, 3(2), 148-158

[19] E. Dror, C. Avin and Z. Lotker. Fast randomized algorithm for hierarchical clustering in Vehicular Ad-Hoc Networks. Ad Hoc Networking Workshop (Med-Hoc-Net), The 10th IFIP Annual Mediterranean, June 2011,1-8

[20] M.S. Almalag and M. C. Weigle. Using traffic flow for cluster formation in vehicular ad-hoc networks. Local Computer Networks (LCN), IEEE 35th Conference, October 2010, 631-636

[21] SUMO – Simulation of Urban Mobility