CVI: Connected Vehicle Infrastructure for ITS

Agop Koulakezian
School of Electrical and Computer Engineering
University of Toronto, Toronto, Canada
Email: agop.koulakezian@utoronto.ca

Alberto Leon-Garcia
School of Electrical and Computer Engineering
University of Toronto, Toronto, Canada
Email: alberto.leongarcia@utoronto.ca

***Abstract*—Intelligent Transportation Systems (ITS) aggregate, analyze and display geographic- and temporal-specific sensor information to reduce congestion while promoting safety. Until now, the coverage and potential of ITS have been restricted by the excessive cost of deploying the required road sensor and communications infrastructure. Our solution to this problem is focused on a novel integrated ITS Network Architecture where vehicles are the main infrastructure in the network. We propose The Connected Vehicle Infrastructure (CVI) for Intelligent Transportation Systems (ITS), where vehicles adapt to mobility changes to form stable vehicular clusters using a Network Criticality-based algorithm we have developed. Further, they build on their clusters to form more stable Mobile Networks, as part of the ITS network. Simulation results using NS-2 show that CVI clustering provides more stable clusters, lower handoffs, higher resilience to errors and better connectivity than popular density-based vehicle clustering methods. In addition, the overhead analysis of CVI shows that it achieves reasonable overhead compared to common clustering algorithms.**

I. Introduction

Modern vehicles include advanced sensor technologies that can provide real-time local information about current road conditions, such as congestion, vehicle speeds, and weather conditions. Intelligent Transportation Systems (ITS) can aggregate and analyze this rich information in order to provide timely local feedback to drivers through public service broadcasts and location-based multicasts. This leads to considerable reduction of congestion and travel time, in addition to notable reduction of greenhouse gas emissions caused by congestion.

In order to provide real-time local-oriented feedback to its users, the ITS Provider network should not only provide a means of access of its users to its central network, but should also support local networks that include intelligent aggregating and processing functions. Such local networks can help correlate events to identify threats and filter out irrelevant events in the continuous vehicular sensor data, consequently saving the bandwidth resources of the ITS central network.

Building on concepts of Mobile IP [1], NEtwork MObility (NEMO) [2] allows a mobile node, called a Mobile Router (MR), to act as the router for nodes moving together as one entity. The MR performs handoff on behalf of these moving nodes, called Mobile Network Nodes (MNNs) and running only Mobile IP, keeping them connected to the Internet with the same IP address and reducing their handoff latency [2].

Being designed for mobile networks having single-hop connectivity to a network infrastructure, such as trains, NEMO alone cannot provide connectivity over multi-hop, intermittent access to the network infrastructure, unless coupled with a Vehicular Ad-Hoc Network (VANET) routing protocol [2], [3]. For this reason, the network architectures presented in [4] and [5] depend on both Network Mobility for providing reachability for a mobile network (also called NEtwork-that-MOves or NEMO) and on VANET routing for handling communication between vehicles and dedicated fixed Road-Side Units (RSUs). These designs require each vehicle to contain a NEMO-capable MR device (based on [2]) and an RSU to be installed at least every two miles to maintain network connectivity. However, the excessive cost of installing and maintaining such RSUs and the need to install NEMO-capable devices in each car cast a shadow on the feasibility, scalability and widespread deployment of VANETs for ITS.

Contribution: We propose The Connected Vehicle Infrastructure (CVI) for ITS, where vehicles are the main infrastructure in the network (Fig. 1). Through beaconing, vehicles take advantage of their estimable mobility and adaptively group themselves into stable clusters of low relative mobility, taking roles of Cluster Member (CM) and Cluster Head (CH) (Fig. 1). CVI considers vehicles as being MNNs, rather than MRs in a NEMO (Fig. 1), providing intelligent aggregating capabilities between vehicle clusters, reducing infrastructure cost and mitigating the overhead of accessing the central network for each vehicle sensor update or topology change. Moreover, CVI provides extended network connectivity by dynamically building on vehicle clusters to form more stable NEMOs, using trucks and 'good-acting' cars (Vehicle E in Fig. 1) as MRs (intelligent local ITS agents). By sharing information between these 'moving infrastructure' elements and predicting topology changes, CVI ensures that NEMO elements move as a unit for long intervals (as required by [2]) and handover between NEMOs is minimized. This leads to efficient node management in terms of addressing scheme, routing and load balancing between NEMOs and clusters.

Since CVI requires a robust and adaptive clustering algorithm, we have developed a novel VANET clustering algorithm based on the Network Criticality metric [6]. Network criticality is a metric that measures the robustness in a network by considering the effects of environmental changes such as load or topology. It is defined as the random-walk betweenness of a node divided by its weight, which is the sum of its incident link weights, thus it measures the robustness of each node with respect to the network [6]. In this paper, we have introduced a localized version of network criticality for a VANET node,

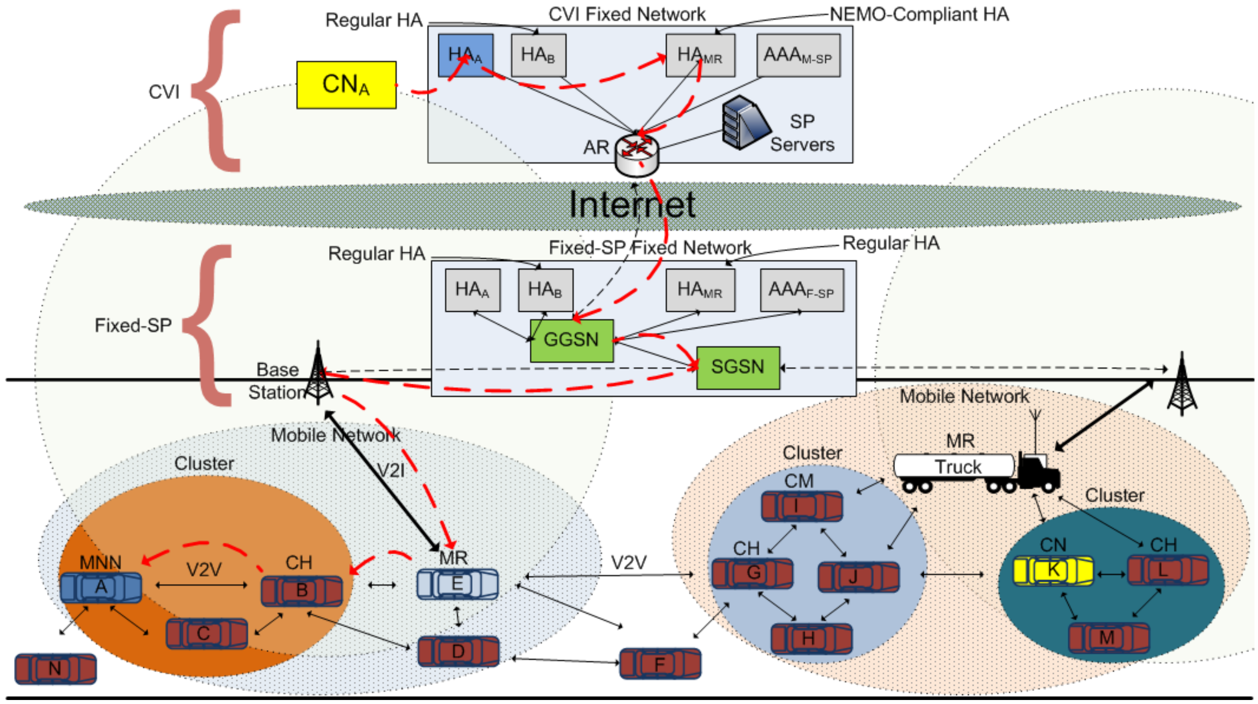


Fig. 1: CVI Network Architecture

which can potentially be promoted to become a clusterhead (if having the lowest criticality value among neighbors). Unlike the metrics of density [7], affinity [8], and relative mobility [9], Criticality adapts robustly to network topology changes in the intense mobile environment of VANETs, where the connections of communication links are short lived. Therefore, it leads to a stable clustering algorithm without the need for time averaging of subsequent values.

The rest of this paper is organized as follows. Sections II and III present the CVI Architecture and its beaconing, maintenance and clustering algorithms, along with their overhead analysis. Section IV presents the simulation results using the NS-2 simulator. We provide concluding remarks in Section V.

II. CVI Network Architecture

Fig. 1 illustrates the proposed Network Architecture. While the Fixed Service Provider (Fixed-SP) is an existent Internet access provider that controls base stations and offers internet access through a GSM network (as shown) or LTE network (in the near future), the CVI Provider (Department of Transportation, Municipality, or a Service Provider) manages moving Infrastructure (trucks and Cars) in the vehicular network, along with Home Agents and various servers in its fixed network. It offers ITS services to subscribers located in the vehicular network and uses the services of the Fixed-SP to access the Internet and reach its fixed network.

Vehicles moving on a highway are grouped into clusters each with a CH and clusters are grouped in NEMOs each with a MR (Fig. 1). As long as a vehicle is within the same Mobile Network, it maintains its IP address and its sessions. We assume that vehicles and trucks both have Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication capabilities, namely that they possess GSM/LTE (V2I) and IEEE 802.11 (V2V) communication capabilities, with ranges of around 1000m and 250m respectively, but only turn on their GSM/LTE antennas when acting as a MR. The sensors within a vehicle are connected to the network through a NAT. This ensures that several sensors in one vehicle use one external IP address (the vehicle IP address) and removes the overhead of using Nested NEMOs (NEMO within a NEMO) [2]. As shown in Fig. 1, a vehicle A can have two types of Correspondent Nodes (CNs), namely a fixed CN (CN_A) acting as a traffic web server, or a mobile CN (vehicle K) sending an accident report to Vehicle A for example.

Our aim is to have vehicles having long intervals in a NEMO, maintaining their session continuity while aggregating their sensor data. We also aim to have load balancing between clusters using a low number of CHs. Hence, we target forming NEMOs with inter-NEMO stability, typically of a few clusters of vehicles, being a few hops away from the MR and having low relative velocities within the NEMOs. Thus, the algorithms presented in Section III aim to provide high network connectivity, low CH change rate and high cluster size.

III. CVI Algorithms

The proposed CVI algorithms use the robust metric of Network Criticality to cluster vehicles. Each node (vehicle) in the network independently computes the link criticalities [6] between itself and each of its 1-hop neighbors using Link Expiration Time [10] as the weight of robustness. Next, it calculates its own (node) criticality as the sum of its link criticalities, transmits it to its neighbors, and makes a clustering decision based on its local view of the network (received criticality values). Based on the current role of a node (CM, CH, or MR), this clustering decision could be an attempt to join a cluster (sent to a CH), join a NEMO (sent to a MR), or become a MR (sent to the CVI through a base station). Alternatively, it could be a response to a cluster join request (if CH), or to a NEMO join request (if MR).

A. Automatic Neighbor Relation Setup

Every node *i* will maintain a neighbor list, $\mathrm{N_i}$, which has a neighbor entry $\mathrm{N_i^j}$ for every neighbor 1 or 2 hops away. This neighbor list is populated by the periodic messages sent to node *i* from these neighbors. Every node *j* will periodically broadcast a HELLO beacon containing its ID ($\mathrm{ID_j}$), position vector ($(\mathrm{x,y})_\mathrm{j}$), velocity vector ($(\mathrm{v_x,v_y})_\mathrm{j}$), criticality ($\tau_\mathrm{j}$), CH ($\mathrm{CH_j}$), MR ($\mathrm{MR_j}$) and cluster size ($\mathrm{CS_j}$). The hello broadcast period is defined as $\mathrm{T_H}$, typically around 1sec. Upon reception of a HELLO beacon from node *j*, node *i* updates $\mathrm{N_i}$ with the new information. In addition, it computes the time that node *j*'s last message expires ($\mathrm{texpire_j}$) and adds it to $\mathrm{N_i^j}$. Next, it forwards the beacon if TTL>0.

B. Cluster Maintenance

In order to dynamically adapt to mobility changes, node *i* runs every $\mathrm{T_H}$ seconds the PURGE algorithm, which is shown in Algorithm 1 and works as follows. All CMs, CHs and MRs periodically check whether they have any expired neighbors that need to be purged (lines 3, 13 and 26). In addition, a MR *i* checks if there is any other MR *j* close to it (line 6) and considers dropping its MR status (line 7): If the situation is persistent after several checks and $\tau_\mathrm{j} < \tau_\mathrm{i}$ (*j* has a better τ), has more NEMO Members, and has enough room for node *i*'s NEMO Members, *i* will cease its role as MR. FutureDist(i,j) (lines 6 and 19) is defined as the estimated distance between nodes *i* and *j* using $(\mathrm{x,y})$ and $(\mathrm{v_x,v_y})$ of each. The distance thresholds of 800m and 450m are used to ensure large clusters and NEMOs assuming rural area coverage. A CH follows a similar procedure by checking if there is another CH *j* close to it (line 20). However, a CH *i* ensures that CH *j* belongs to the same NEMO also, or else it does not cease its role as CH. This ensures that bad acting CHs will not remain indefinitely, and small clusters will merge together to improve load balancing, yet ensure low handoffs between NEMOs (changing a MR). Moreover, a CH checks whether its MR is present and is still a MR (line 16) and a CM checks whether its CH is present, is still a CH and has the same MR as before (line 29). Note that our algorithms do not assume synchronization and each vehicle can run them independently from one another.

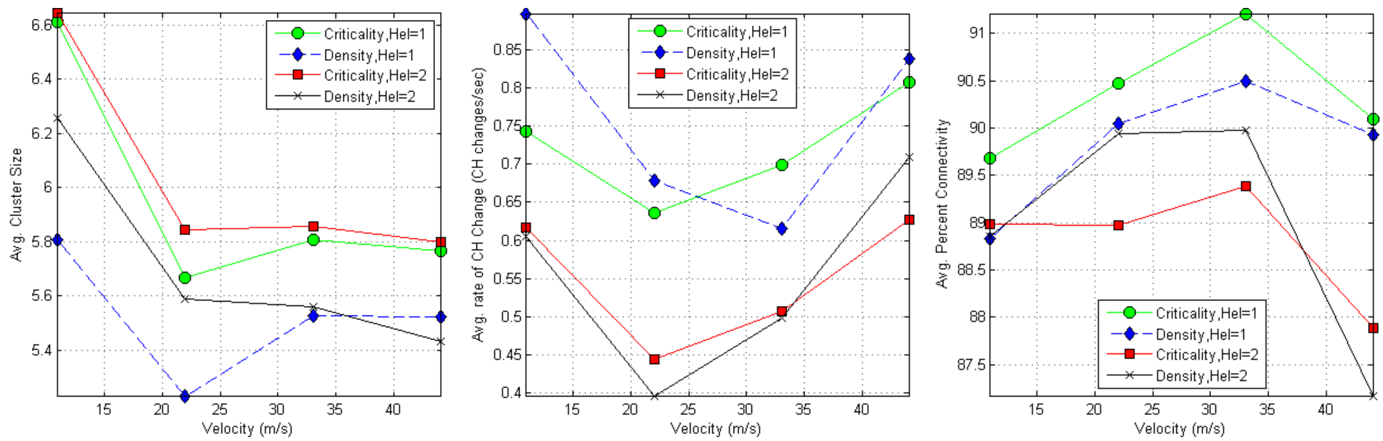
1: **if** $\mathrm{ID = MR}$ **then**
2:     **for all** $\mathrm{N_i^j \in N_i}$ **do**
3:         **if** $\mathrm{texpire_j < t}$ **then**
4:             Purge $\mathrm{N_i^j}$ from $\mathrm{N_i}$
5:         **end if**
6:         **if** $\mathrm{CS_i < Threshold_{MR}}$ & $\mathrm{MR_j = ID_j}$ & FutureDist(i,j) < 800 **then**
7:             *Consider Dropping MR Status*
8:         **end if**
9:     **end for**
10: **else if** $\mathrm{ID = CH}$ **then**
11:     $\mathrm{Lost_{MR} \leftarrow 1}$
12:     **for all** $\mathrm{N_i^j \in N_i}$ **do**
13:         **if** $\mathrm{texpire_j < t}$ **then**
14:             Purge $\mathrm{N_i^j}$ from $\mathrm{N_i}$
15:         **end if**
16:         **if** $\mathrm{MR_i = ID_j}$ & $\mathrm{MR_j = ID_j}$ **then**
17:             $\mathrm{Lost_{MR} \leftarrow 0}$
18:         **end if**
19:         **if** $\mathrm{CS_i < Threshold_{CH}}$ & $\mathrm{CH_j = ID_j}$ & FutureDist(i,j) < 450 **then**
20:             *Consider Dropping CH Status*
21:         **end if**
22:     **end for**
23: **else**
24:     $\mathrm{Lost_{CH} \leftarrow 1}$
25:     **for all** $\mathrm{N_i^j \in N_i}$ **do**
26:         **if** $\mathrm{texpire_j < t}$ **then**
27:             Purge $\mathrm{N_i^j}$ from $\mathrm{N_i}$
28:         **end if**
29:         **if** $\mathrm{CH_i = ID_j}$ & $\mathrm{CH_j = ID_j}$ & $\mathrm{MR_i = MR_j}$ **then**
30:             $\mathrm{Lost_{CH} \leftarrow 0}$
31:         **end if**
32:     **end for**
33: **end if**

Algorithm 1: Purge (i,t)

C. Cluster Formation

Every $\mathrm{T_H}$ seconds, node *i* runs the following cluster formation algorithm: In case it has $\mathrm{Lost_{CH}}$=1 (in Algorithm 1), it starts looking for a new CH in a possible set in $\mathrm{N_i}$, choosing the one with the lowest τ_j. The first possible set is made of CHs that have $\mathrm{MR_j = MR_i}$. The next possible set is made of CHs that have any MR and the last possible set is made of the remaining CHs in $\mathrm{N_i}$. The last case only fails if a node does not have any non-full clusters around it, or it has a better τ than all possible CHs. In this case, node *i* sets itself as a CH of a new cluster, aiming for other nodes to try to join its cluster. If node *i* finds a possible CH *j*, whose cluster it would like to join, it sends *j* a *Cluster Join Request* that gets forwarded by its neighbors (if needed) to reach node *j*. If successfully received, node *i* gets an accept/reject decision for its request from node *j* based on $\mathrm{CS_j}$ and FutureDist(i,j). In a similar (but simpler) fashion, CH *i* looks for a new MR in case it has $\mathrm{Lost_{MR}}$=1 and sends it a *NEMO Join Request*. In case persistent attempts to join a NEMO fail (this situation would be similar to a Truck or car, which is acting as a MR, leaving the highway), a CH sends a *Become MR Request* to the CVI through a base station and receives an accept/reject decision based on the existence of MRs close to it.

The CVI algorithms use *richer* information from the CVI network, which has a better view of the network than the local view of each node. This helps intelligently react to topology



(a) Cluster Size (b) Clusterhead Changes (c) Network Connectivity

Fig. 2: CVI Mobility Performance

changes with τ and future mobility estimation, creating large clusters and providing high network connectivity.

D. Overhead Analysis

In addition to forming stable clusters, the CVI Algorithms aim to reduce the number of messages sent through the network, that is the messaging overhead. This overhead is mostly due to the periodic Hello messages for automatic neighbor relation setup (Secion III-A) , as well as some event-based messages for cluster maintenance (Secion III-B) and formation (Secion III-C). All of these messages include IP and MAC headers, position and velocity information, the criticality value, in addition to the remaining elements of the neighbor list, $\mathrm{N_j}$ (for node *j*), as discussed in Secion III-A. Each of these values is assumed to require 4 bytes, except for the cluster size ($\mathrm{CS_j}$), which requires 1 byte. Thus, the size of the CVI message (for node *j*) is:

$$\begin{aligned} size_{message} =& \mathrm{LEN_{IP-HDR}} + \mathrm{LEN_{MAC-HDR}} + (\mathrm{x},\mathrm{y})_\mathrm{j} \\ &+ (\mathrm{v_x},\mathrm{v_y})_\mathrm{j} + \tau_\mathrm{j} + \mathrm{CH_j} + \mathrm{MR_j} + \mathrm{CS_j} \\ =& 20 + 58 + 8 + 8 + 4 + 4 + 4 + 1 \\ =& 107 Bytes/message \end{aligned} \quad (1)$$

While the overhead of periodic Hello messages is independent of the metric used in the CVI algorithm (such as Criticality), the choice of this metric greatly affects the number of event-based messages for cluster formation and maintenance, changing the total message overhead. Moreover, high mobility in vehicles leads to greater forwarding of Hello messages, thereby increasing the message overhead. In Section IV-B, we study the effect of mobility and the choice of the CVI algorithm metric on its messaging overhead.

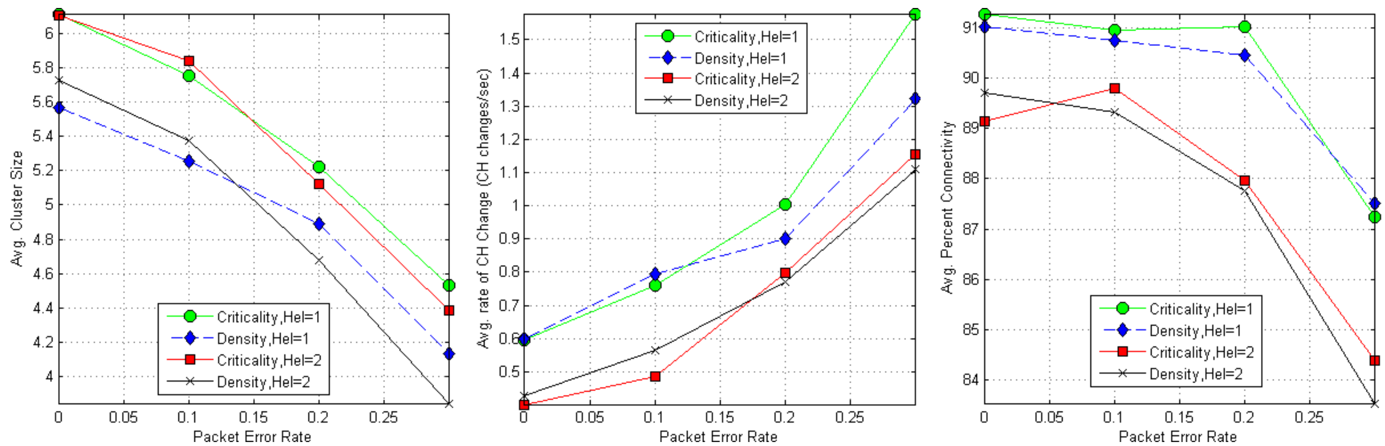
IV. Simulation Results

We have implemented the proposed algorithms in NS2 [11]. All simulations were performed with 400 vehicles on a looped 30 Km, 2-lane highway moving in one direction and including lane-changing, generated with MOVE [12], which is based on the open-source traffic simulator, SUMO [13]. MOVE outputs realistic NS2 traces, which were used in the NS2 simulations. Each simulation ran for 1500s, however only the last 300s were used for performance metric calculations. This was to ensure that all 400 vehicles had successfully entered the highway and reached their respective velocities. All of the simulation results were averaged over 5 different mobility scenarios. We used IEEE 802.11 as the MAC protocol in the NS2 simulations, with the 914MHz Lucent WaveLAN DSSS network card, the two-ray ground propagation model and a radio range of 250m, as is common in VANET literature [8], [9]. We note that the newly developed radio models for V2V communications by some research groups are not public or have yet to be thoroughly verified [11].

The vehicles reach their maximum speed if possible and slow down at the turns or when blocked by slower vehicles (only 2-lanes are used), creating a realistic pattern with both low and high density traffic. To ensure load balancing, CHs have a max cluster size of 15 and MRs have a max cluster size of 60. Note that regular nodes broadcast HELLO beacons with TTL=2, while MRs broadcast HELLO beacons with TTL=4 (for larger coverage) in our simulations.In the typical scenario, vehicles with low relative mobility, such as trucks, would be the MRs. However in our simulations, vehicles which have a NEMO-capable MR device could instead take on this role (we do not include trucks in these simulations). Therefore, we are testing CVI algorithms under very harsh mobility conditions. Simulations with trucks have been left out of this paper, but are presented in [14].

The stability of the Network Criticality-based clustering algorithm (without using NEMO) has been studied in [15], where it provided longer clusterhead durations and lower clusterhead changes than common clustering algorithms such as the Modified Distributed and Mobility-Adaptive Clustering (MDMAC) algorithm [16]. In this paper, we focus on the creation of NEMOs and particularly NEMO-based clustering.

As no clustering algorithm leads to the creation of NEMOs in the literature, we compare the performance of τ relative to



(a) Cluster Size (b) Clusterhead Changes (c) Network Connectivity

Fig. 3: CVI Robustness to Channel Error

density [7] within the realm of the CVI algorithms. For density simulations, each node *i* broadcasts its Density (defined as the number of entries in $\mathrm{N_i}$), instead of τ_i to its neighbors. Clustering and maintenance decisions are made using the inverse of Density, rather than τ. Thus, nodes with high Density are more likely to be chosen as CHs. Although vehicles are moving relative to their NEMOs, we show here that the CVI algorithm leads to high network connectivity (connectivity to a MR, thus the Internet), validating the CVI Network Architecture presented in Section II. Moreover, we show that the Criticality metric is suitable for clustering, is at least as good as the density metric in terms of network connectivity and provides larger clusters than Density.

A. *Performance Metrics*

To evaluate the cluster stability, the validity of the CVI Architecture and the overall performance of our algorithm, we use the following metrics:

1) Average Cluster Size: Large cluster sizes are important for efficient caching, management, and load balancing, where the CH is the central controller.
2) Average Rate of Clusterhead Change: This metric is useful since it takes into account both CH duration and the number of clusters formed.
3) Average Network Connectivity: This metric measures the percentage of time a node is connected to a MR, thus to the internet. It accounts for the time a node spends being a member of a NEMO (as a CM or CH) and does not include the time when a node is searching for a MR.

B. *Performance Analysis*

In the first set of simulations, CVI mobility performance is compared using both the Criticality and Density metrics. Using SUMO, highway scenarios are generated with a Gaussian distribution of velocities with means of 11, 22, 33, and 44 m/s. $\mathrm{T_H}$ is also swept from 1 sec to 2 sec in these simulations to self-assess the CVI algorithms (Fig. 2).

Fig. 2-(a) shows that Criticality leads to a significantly higher average cluster size (lower number of clusters) compared to Density. Fig. 2-(b) shows that whenever the CVI algorithms are run with $\mathrm{T_H}$=2s, rather than $\mathrm{T_H}$=1s, the CH changes will be lower (for both Criticality and Density) and this will result in a higher cluster size as seen in Fig. 2-(a). However, this also results in far lower network connectivity as seen in Fig. 2-(c). Therefore, when $\mathrm{T_H}$=2s, the algorithm reacts less often to changes, thus does not adapt fast enough to mobility changes, leading to a lower network connectivity, especially at higher speeds as seen in Fig. 2-(c).

Fig. 2-(c) shows that Criticality leads to a slightly higher average network connectivity than density for $\mathrm{T_H}$=1s. The low network connectivity for 11 m/s and 22 m/s speeds, relative to 33 m/s speeds, is due to the many CHs within the range of each other, leading to running *Consider Dropping CH Status* frequently. However, as Fig. 2-(b) measures the average rate of CH changes/sec for the 400 nodes, the number of CH changes/sec/node is very low for both Criticality and Density. This validates the CVI algorithms presented in section III, including the *Consider Dropping CH Status* function.

Next, we measure the robustness of the CVI algorithms to channel error using both Criticality and Density. We have chosen the speed of 33 m/sec here, as it is the most realistic average speed on a highway worldwide. In these simulations, messages are kept for longer periods (texpire is higher) for both Criticality and Density. The performance results are displayed in Fig. 3, where the Packet Error Rate (PER) is swept from 0 to 0.3 in increments of 0.1. As before, Criticality leads to a higher average cluster size (Fig. 3-(a)) and slightly higher average network connectivity (Fig. 3-(c)) with both values decreasing with the increase in PER. Moreover, Fig. 3-(b) shows that the higher PER leads to an increase in CH changes for both Density and Criticality, as expected. Note that for $\mathrm{T_H}$=1s, both Criticality and Density lead to high average network connectivity upto PER=0.2, which shows the robustness of the CVI algorithms to channel error.

Finally, we study the overhead performance of the CVI

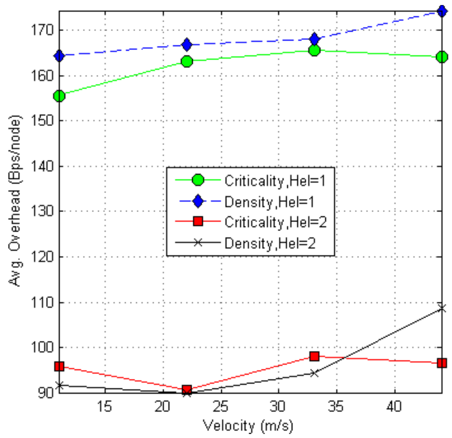


Fig. 4: CVI Overhead Performance

algorithms. The overhead, measured in Bytes/sec/node, is computed by counting the total number of bytes from all periodic and event-based HELLO messages sent for beaconing, maintenance and cluster formation and averaged per every second and every node. The overhead performance of CVI using Criticality and Density are compared in Fig. 4, where Criticality has a slightly lower average overhead than Density, especially for T_H=1s. Because Criticality takes into account the robustness of the vehicular network, it leads to clustering decisions that decrease the number of messages (and thus the messaging overhead) required to maintain the formed clusters.

Similar to Fig. 2-(b), the overhead of T_H=2s is almost half that of T_H=1s as vehicles broadcast their HELLO messages with double the period. However, this gain is at the expense of more event-based HELLO messages and lower network connectivity (Fig. 2-(c) and Fig. 3-(c)). Furthermore, increasing the vehicle mobility leads to a higher forwarding rate for Hello messages, which increases the overhead, as expected. The resulting analysis shows that the CVI algorithms have a reasonable overhead, comparable to that of clustering algorithms in the literature, as studied in [8] for Affinity Propagation clustering [8] (around 170 Bps/node) and MOBIC [17] (around 180 Bps/node).

V. Conclusion

We presented CVI, a novel integrated ITS Network Architecture that exploits vehicle mobility and builds stable vehicular clusters as ITS network infrastructure. Moreover, we presented a novel VANET clustering algorithm that helps nodes dynamically adapt to mobility changes and regroup in order to form stable, long-lasting vehicle clusters and NEMOs, using the robust Network Criticality metric and mobility estimation. This leads to large clusters of vehicles, with low average rate of CH change and high network connectivity, resulting in the efficient aggregation of vehicular sensor data. Furthermore, due to the intelligent maintenance algorithms presented, CVI also achieves high robustness to channel error and reasonable messaging overhead.

Acknowledgments

The authors would like to thank Christine Shea, Weiwei Li and Ali Tizghadam for their support in this work.

References

[1] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775 (Proposed Standard), IETF, Jun. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3775.txt

[2] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963 (Proposed Standard), IETF, Jan. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3963.txt

[3] R. Baldessari, A. Festag, and J. Abeille, "Nemo meets vanet: A deployability analysis of network mobility in vehicular communication," in *Telecommunications, 2007. ITST '07. 7th International Conference on ITS*, june 2007, pp. 1 –6.

[4] R. Baldessari, W. Zhang, A. Festag, and L. Le, "A manet-centric solution for the application of nemo in vanet using geographic routing," in *TridentCom*, 2008, pp. 1–7.

[5] M. Tsukada, O. Mehani, and T. Ernst, "Simultaneous usage of nemo and manet for vehicular communication," in *TridentCom*, 2008, pp. 1–8.

[6] A. Tizghadam and A. Leon-Garcia, "Survival value of communication networks," in *INFOCOM Workshops 2009, IEEE*, april 2009, pp. 1 –6.

[7] S. Kuklinski and G. Wolny, "Density based clustering algorithm for VANETs," in *TridentCom*, Washington, DC, 2009, pp. 1–6.

[8] C. Shea, "Aprove: A stable and robust vanet clustering scheme using affinity propagation," Master's of Applied Science, University of Toronto, Department of Electrical and Computer Engineering, 2009.

[9] S. Gopalaswamy, "Dynamic clustering protocol based on relative speed in mobile ad hoc networks for intelligent vehicles," MSc, Auburn University, Dept of Computer Science and Software Engineering, 2007.

[10] W. Su, S. ju Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," in *International Journal of Network Management*, 2000, pp. 3–30.

[11] "The network simulator ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns

[12] F. K. Karnadi, Z. H. Mo, and K. chan Lan, "Rapid generation of realistic mobility models for VANET," in *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pp. 2506–2511.

[13] "Sumo: simulation of urban mobility." [Online]. Available: http://www.sumo.sourceforge.net

[14] A. Koulakezian, "Aspire: Adaptive service provider infrastructure for vanets," Master's of Applied Science, University of Toronto, Department of Electrical and Computer Engineering, 2011.

[15] W. Li, "Network clustering in vehicular communication networks," Master's of Applied Science, University of Toronto, Department of Electrical and Computer Engineering, 2011.

[16] G. Wolny, "Modified dmac clustering algorithm for vanets," in *Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference on*, oct. 2008, pp. 268 –273.

[17] P. Basu, N. Khan, and T. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Distributed Computing Systems Workshop, 2001 International Conference on*, apr 2001, pp. 413 –418.