CS224W 2017 Project Final Report:

Clustering-based Scalable Evolving Graphs for Vehicular Ad Hoc Networks

Masatoshi Tanida (mtanida@outlook.com) and David Wugofski (dwugo@stanford.edu)

***Abstract*—Message routing in vehicular ad hoc networks (VANETs) pose a particular technical challenge due to their highly dynamic nature. Eiza and Ni [1] have proposed an evolving graph approach to route messages in VANETs both reliably and efficiently. Evolving graphs, however, are resource intensive and do not scale well as the density of vehicles in the network increases. In this paper, we present a hybrid approach that combines evolving graphs with node clustering techniques to create scalable evolving graph models that are independent of node density. We evaluate our approach on a large-scale geolocation dataset obtained from taxis in Beijing, China.**

I. Introduction

Wireless communication between vehicles is a well-researched but ultimately unresolved field of exploration. Due to the high mobility of these vehicles, it is difficult to establish reliable networks for any length of time. As such, models for wireless communication between vehicles consistently include a reliance on ad-hoc networking, where the network architecture is determined by the vehicles themselves in the environment rather than by a central authority, with vehicles passing data packets between themselves in order to reach their intended destinations.

The difficulty in establishing these vehicle ad-hoc networks (VANETs) lies in determining the protocols vehicles should use to determine routing paths to send messages along. Currently, the closest similar model to VANETs are mobile ad-hoc networks (MANETs). However, the two problems present different sets of advantages and disadvantages. MANETs change much more slowly than VANETs, as MANETS are typically limited by human mobility. However vehicle movement can be much more easily predicted, and vehicles typically allow for greater power output and computational complexity than mobile devices.

Many approaches have been considered to try leveraging these advantages to solve the VANET routing problem. This paper explores an approach discussed by Eiza and Ni [1] based on Evolving Graph Theory. In [1], the VANET is viewed as a graph, with nodes which connect and disconnect to and from each other over time.

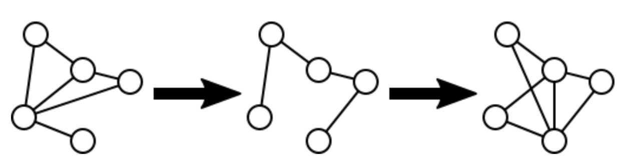

Fig. 1: Depiction of an Evolving Graph

The evolving graph reliable ad hoc on-demand distance vector routing protocol (EG-RAODV) method proposed by [1] attempts to assign edge *reliability* measures to attempt to predict which paths will remain open during transmission.

The work done by [1] was based on an exploration of vehicles traversing a highway. In addition to basing their results on somewhat unrealistic environments, the approach they used was highly inefficient for large, dense environments.

In this paper, we simulate the performance on the EG-RAODV algorithm on a real data set of taxi cabs in Beijing. Additionally, we present a hybrid algorithm that combines a heuristic-based clustering algorithm with EG-RAODV that allows EG-RAODV to scale with the vehicle density. We simulate the performance on the hybrid algorithm on the same taxi data set.

The rest of the paper is organized as follows. Section (2) describes past works relevant to our project. Section (3) describes the data set that we use as well as the preprocessing steps we perform on it. Section (4) details how we apply the EG-RAODV algorithm to our data set. Section (5) details the clustering algorithm we use. Section (6) talks briefly about the hybrid clustering-based EG-RAODV approach. Section (7) presents the results of our simulations. Finally, Section (8) concludes the paper.

II. Previous Work

Much of the previous mathematical exploration and rigor of this paper is based on the discussion provided by [1]. However, their exploration has left room for further investigation, which we expand upon.

A. The EG-RAODV Algorithm

Eiza and Ni [1] formally combined the concepts of Evolving Graph Theory with VANET routing. In the Evolving Graph Model, connections between nodes form and break over time, making it possible for a path to exist between two nodes at one point, and not exist the next.

In this setting, it is crucial to anticipate changes in network topology before they occur, otherwise a vehicle may attempt a routing path which will fail before the packet has a chance to arrive at its destination.

B. Link reliability

In order to anticipate link breakages, [1] uses *link reliability*, which expresses the probability a link is going to fail during communication. To model this probability, [1] assumes vehicle velocity v is distributed according to a *normal distribution*. Thus the relative velocity between any two vehicles, Δv, is also normal, with mean $\mu_{\Delta v}$ and standard deviation $\sigma_{\Delta v}$.

If the maximum radio communication radius of a vehicle is H, the reliability $r(l)$ of link l connecting two nodes i and j can be thought of as the probability that the two vehicles will remain within a diameter of $2H$ within each other ($-H$ and $+H$ distances apart).

Since the probability that the vehicles remain in this region is based on their relative velocity, we have that the link reliability at time t is:

$$r_t(l) = \begin{cases} \rho(t) - \rho(t + T_p) & T_p > 0 \\ 0 & \text{else} \end{cases} \tag{1}$$

$$\rho(t) = \text{Erf}\left[\frac{\left(\frac{2H}{t} - \mu_{\Delta v}\right)}{\sigma_{\Delta v}\sqrt{2}}\right] \tag{2}$$

$$T_p = \frac{H - \sqrt{(y_i - y_i)^2 + (x_i - x_i)^2}}{|\Delta v_{i,j}|} \tag{3}$$

$$\text{Erf}(\tau) = \frac{2}{\sqrt{\pi}} \int_0^{\tau} e^{-t^2} dt \tag{4}$$

In VANETs, multiple potential routes can exist between vehicles. As well, the most reliable route is not necessarily the most direct, since a less direct route may take advantage of more reliable communication links. We denote a route (or path) P traversing a set of links as $P = (l_1, l_2, ..., l_k)$. The route reliability can be calculated as:

$$R_t(P) = \prod_{i=1}^{k} r_t(l_i) \tag{5}$$

We can therefore calculate the most reliable journey (MRJ) from a source node s to a destination d as

$$\text{MRJ}_t(s, d) = \max_{\{P: s \to d\}} R_t(P) \tag{6}$$

where $\{P : s \to d\}$ is the set of all paths from s to d.

C. The EG-Dijkstra Algorithm

Once link reliability is established, the question becomes how to find the MRJ efficiently. Building off the Dijkstra algorithm for finding the shortest path between two nodes, [1] develop the Evolving Graph (EG-)Dijkstra algorithm. The algorithm is outlined in Algorithm 1.

The EG-RAODV algorithm uses this MRJ to build a routing request to send data to another vehicle in the network. Unfortunately, for dense networks, the Dijkstra algorithm is very inefficient. We seek to improve its performance by incorporating clustering, which would enable us to calculate the MRJs faster, hopefully without significantly compromising the reliability of the route.

Algorithm 1 EG-Dijkstra

Input s_r
Output array RG
$RG(s_r) \leftarrow 1$
$RG(u! = s_r) \leftarrow \emptyset$
$Q \leftarrow \{s_r\}$
while Q not empty **do**
    $x \leftarrow$ the vehicle with the highest reliability in Q
    Mark x as visited
    for v neighbor of x **do**
        if $RG(v)! = \emptyset$ **then**
            $RG(v) \leftarrow r_t(l_x^v) \times RG(x)$
        else
            $RG(v) \leftarrow \max(RG(v), r_t(l_x^v) \times RG(x))$
        if v not previously visited **then**
            $Q \leftarrow Q \cup \{v\}$
    $Q \leftarrow Q \setminus \{x\}$

D. Clustering

The EG-RAODV algorithm described in the previous section present two challenges when applied to real VANET networks. First, it requires vehicles to have global knowledge of the entire network. In order to run the algorithm, each vehicle must have up-to-date information about every other vehicle in the network. In the absence of a centralized network, this can only be accomplished with significant communications overhead. The second challenge is that the run time of the algorithm increases rapidly with the number of nodes in the network. Because of these issues, we propose a hybrid approach that groups vehicles into clusters and applies the EG-RAODV algorithm to the clusters rather than all of the individual nodes in the network.

Node clustering in MANETs and VANETs is an active research area. Countless approaches have been proposed but the basic idea remains the same - to transform a dense graph with many nodes into a smaller, more manageable graph of super-nodes or *clusters*. Finding the optimal clustering configuration reduces to the set-cover problem which is known to be NP-complete, and is therefore intractable.

Moreover, common clustering algorithms found in static networks are less practical in decentralized dynamic networks for the same reason that makes EG-RAODV problematic. That is, most algorithms require full global knowledge of the network. For this reason, the vast majority of clustering approaches that have been proposed for VANETs involve local, heuristic-based algorithms. A common approach involves having each individual node compute its own suitability for becoming a cluster head (CH). Then, the node with the highest suitability score is elected as the CH and its neighbors can elect to join as a member of its cluster. The vast majority of these approaches use a simple weighted combination of its features (such as the node's degree, speed, and direction) to compute the the CH suitability score.

Attempts have been made to compare the various clustering approaches in [2] and [3] but few present empirical results on real world data sets, choosing instead to use simple simulations. Moreover, we are not aware of any previous works that present results obtained from real VANET data.

III. Data Set

In order to more accurately determine the efficiency of our algorithm we use a real-world data set from [4], [5], [6]. This data set tracks the location of 10,357 taxi cabs as they traverse Beijing over the course of a week. Each data point collected records the id number of a taxi, the time the sample was taken, and the latitude and longitude of the taxi cab at that time.

Using this data set, we obtain snapshots of vehicle locations at various points in time. We model the radio-link connectivity as a simple function of distance. Specifically, an edge exists between two vehicles if their Euclidean distance is within some specified range. Using these taxi cabs as our vehicles, we can simulate the conditions of a VANET in a more realistic, complicated scenario. Figure 2 plots the geolocations of all of the taxis in the data set at a particular time.

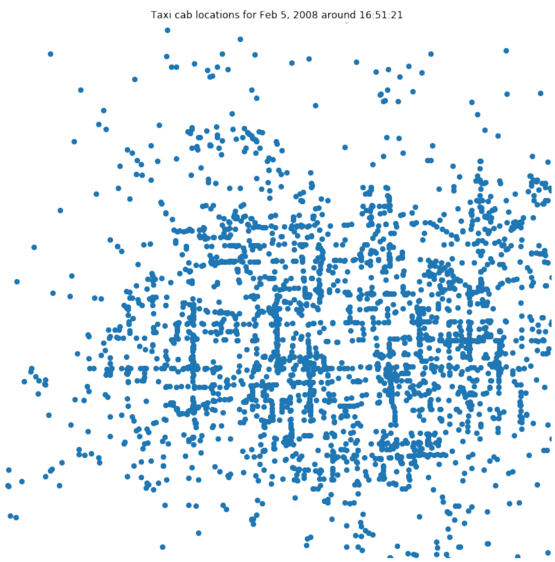

Fig. 2: Taxi cab locations for Feb 5, 2008 around 16:51:21

A. Analysis of Data Set

Figure 3 plots the average node degree (computed for a subset of approximate 4000 nodes) as a function of the transmission radius, H. The plot shows that the function is quadratic with respect to the radius. This is precisely what we expect since the area of a circle is quadratic with respect to its radius. Similarly, Figure 4 plots the average clustering coefficient as a function H. The clustering coefficient approaches 1.0 at a rate that appears logarithmic in H.

Figure 5 plots the histogram of node speeds in meters-per-second (rounded to the nearest integer) for the same data set. Surprisingly, nearly a third of the nodes have a a speed of 0 - that is, they were effectively stationary. Most of the vehicles that were non-stationary were driving at moderate speeds appropriate for an urban streets. A relatively small number of vehicles are moving at highway speeds.

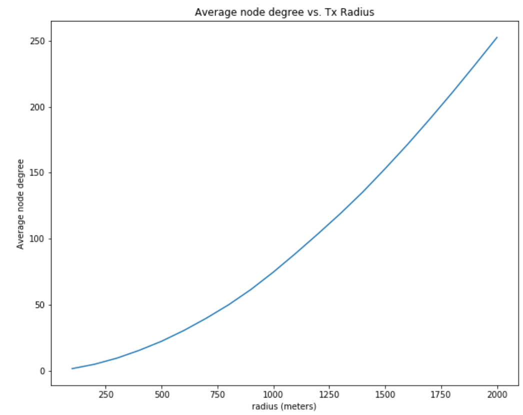

Fig. 3: Average node degree as a function of transmit radius H

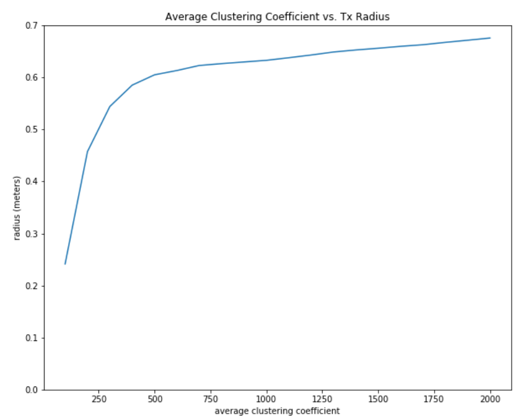

Fig. 4: Average clustering coefficient as a function of transmit radius H

B. Preprocessing

The taxi data set presented a number of challenges. As noted earlier, the data set spans an entire week which is far more than needed for simulation purposes. Hence we take a short slice of time (a duration of 120 seconds) and run our simulations starting at t=10s in that time slice.

Additionally, the location sampling for the data set is sparse, with many samples being taken at intervals of several minutes. As a result, we have had to interpolate positions for our taxis. Since the intervals between samplings are inconsistent, we have shifted taxi cab data around in time in order to focus on the most data-rich intervals. That is, we take data from different times for each taxi in order to simulate a date set that has a denser sample rate. We assume that this does not affect the fidelity of the data since taxis are not likely to have significant temporal route patterns in a dense urban city like Beijing.

The data set also contains incorrect and invalid data, as well as taxis with very few or even zero samples. We discard these

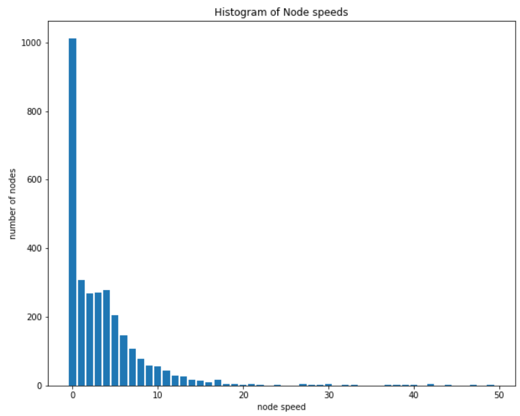

Fig. 5: Histogram of node speeds in meters-per-second computed for a snapshot in data set

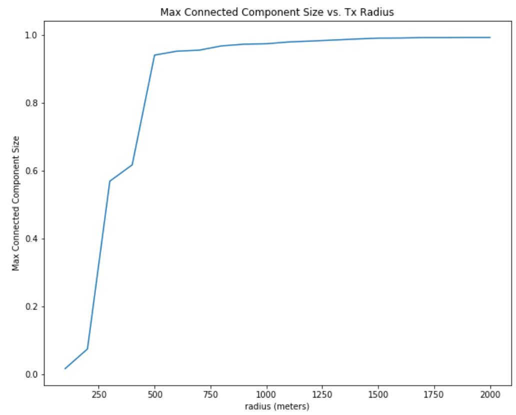

Fig. 6: Proportion of total network covered by the largest connected component as a function of transmit radius H

data points from the data set. We performed additional filtering to remove taxis that were helpful such as taxis that have too few samples or taxis that were located far from the city center. This left us with a date set consisting of 3017 taxis. The rest of the paper, including the simulation results we present, refer to this data set.

IV. EG-RAODV Implementation

From the preprocessed data set, we create graphs for each time step where each node in the graph represents a taxi and presence of an edge between two nodes represents their ability to communicate. Specifically, if two cars are within transmission range, an edge exists between them. Otherwise, they are left unconnected.

There is no concrete answer for the correct transmission radius H should be. Transmission range for wireless telematics modules (DSRC) typical in many vehicles can vary from several kilometers on an open stretch of highway with line-of-sight view to under 100 meters in a very dense urban environment. Moreover, the range depends on the transmit power, which in most cases, is variable. Therefore, we examine our network to find an appropriate value for H.

Since we are interested in routing messages between arbitrary nodes in the network, it is vital that the network is largely connected when we run our simulations. Figure 6 plots the proportion of the total taxi network covered by the largest connected component as a function of H. This is computed for time step $t = 10s$. As the plot shows, the proportion of nodes covered by the largest connected component rises above 0.9 for a value of H just below 500m. Hence, we explore values of H near this range.

A. *Link Reliabilities*

After extracting the taxi locations, we move on to calculating their link reliabilities. Using their location data, we extract the velocities of the cars, and thus the relative velocities. These data points allow us calculate link reliabilities for different vehicles.

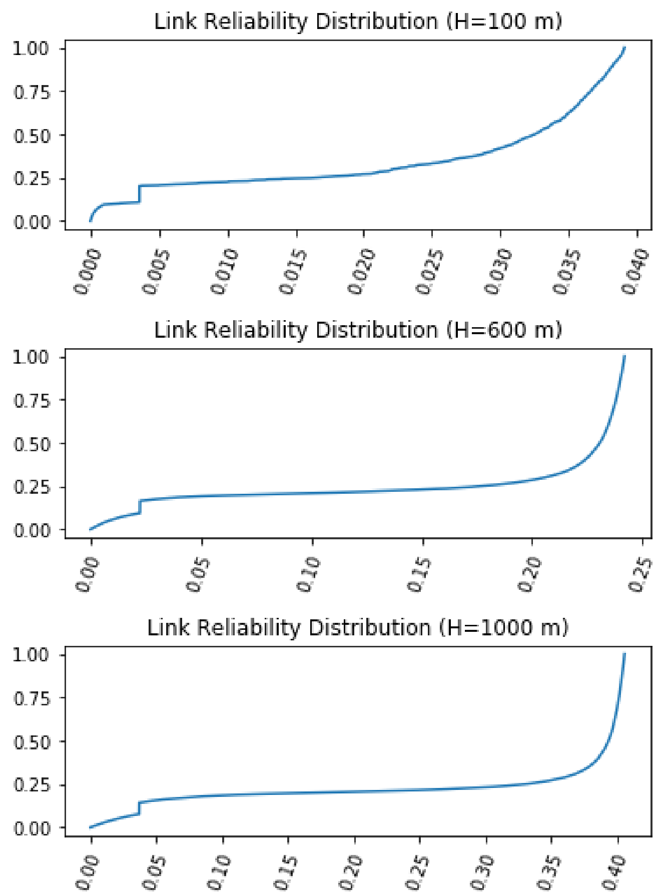

Fig. 7: Link reliability distribution for $H = 100, 600, 1000$ m, respectively

Figure 7 demonstrates an outline of the empirical cumulative density function (CDF) of link reliability for various values of distance. As our results show, changing the communication radius can have a significant effect on not only the number of edges which can form, but how many of those edges fall into a more/less reliable category.

When the maximum communication radius is only 100 m, the empirical CDF for the link reliability is convex. This implies that more than half of the links experience link reliability less than half optimal. As the communication radius goes to 300 and 1,000 m, the function becomes more concave, allowing a greater portion of the links to be close to maximally reliable.

B. MRJ Calculation

While the EG-Dijkstra algorithm may be as efficient as one can hope for when asked to search for the MRJ between two nodes, our project has been implemented in Python, which has created bottlenecks due to computational overhead.

As [1] noted - and rejected as outside its scope - dense VANETs do not mix well with the EG-Dijkstra algorithm. We have found, however, that our taxi data set is not particularly dense, thus we are able to run the algorithm on the full data set on the order of several seconds per MRJ calculation. The computation time varies for different values of H. Timing results are presented in Section (7).

V. Clustering Algorithm

As noted earlier, well known algorithms for static networks are not practical in VANETs because it requires nodes to maintain up-to-date information about every other node in the network. This imposes significant communication overhead on the network. Thus, local, heuristic based clustering approaches are preferred.

We propose a clustering algorithm that is a variation of common VANET clustering heuristics. The pseudocode is as follows:

Algorithm 2 GetClusters

Input Nodes
Output CHList
for all nodes u in Nodes **do**
    compute $w(u)$
while not converged **do**
    for all nodes u in Nodes **do**
        if any neighbor of u is a CH **then**
            Assign u to the CH with highest w score
        else
            if u has highest w among its neighbors that are not yet in a cluster themselves **then**
                Promote u to be a CH
return CHList

In Algorithm 2, each node u computes its CH suitability score $w(u)$ and broadcasts it to its neighbors. If a node has the highest suitability score among its neighbors that do not yet belong to a cluster themselves, then it becomes a CH. This is repeated until convergence - that is, until every node belongs to a cluster. In our experiments, node convergence usually occurs within 10 iterations. We use a simple function to compute the suitability score $w(u)$:

$$w(u) = \sum_{v \in (nbrs)(u)} 1 + d_{uv}/H \tag{7}$$

In the equation, $\text{nbrs}(u)$ denotes the set of one-hop neighbors of node u and d_{uv} is the distance between node the nodes u and v. The 1 component in the summation gives greater score to nodes with many neighbors. The d_{uv}/H component gives favors nodes who have nodes that aren't too close. This is a heuristic that we arrived at through experiments.

Since every vehicle is necessarily GPS-enabled, every node has access to GPS time. Hence we propose a simple, time-interval based mechanism for triggering a cluster reselection. However, we do not investigate good values for this time-interval in this paper.

The advantage to this kind of approach is that nodes need only have knowledge of their local neighbors. This can be achieved by having nodes periodically broadcast *HELLO* messages with information about its geolocation, speed, direction, and so on. CH nodes must have knowledge of all other CH nodes, however, which results in some communication overhead. Mechanism exist for coordinating and disseminating update messages in existing wireless networks, however, we consider the topic beyond the scope of our project and we will not discuss it further.

VI. Hybrid EG-RAODV

The hybrid EG-RAODV approach combines clustering with the EG-RAODV approach. Rather than creating an evolving graph network and running EG-Dijkstra on the full taxi set, we use the subset of taxis comprised of the current cluster head (CH) nodes. Consequently, the MRJ path that we compute between an arbitrary source and destination node is the path from the source node's CH to the destination node's CH. Note that the source and destination nodes themselves may be CHs.

While the MRJ is a path along CHs, the messages can be transmitted through non CH nodes. Specifically, when passing a messages from the current cluster to a target cluster, the current cluster's CH can broadcast the message to the members of the cluster, and those members can subsequently pass the message on to a member of the target cluster. Thus, the CH nodes are not required to be one-hop neighbors.

VII. Simulation and Results

The simulations are set up as follows. For each simulation, we randomly select a pair of nodes for the start and destination node. We limit the selection to nodes in the largest connected component of the network at time t_0 (note, however that this does not ensure that a path will exist from source to destination as the vehicles move over time). Then, given the source and destination nodes, we attempt to generate the MRJ. This step can have two possible results: either the reliability score of the path does not meet a certain threshold, in which case, we mark the simulation as a failure, or it will return the MRJ and its associated reliability score. If we obtain an MRJ, we run the simulation. Algorithm 3 is the pseudocode for the simulation. Currently, the algorithm is entirely deterministic, but we will explore the possibility of making it probabilistic for the final report.

Algorithm 3 RunSim

Input MRJ
Output $\{1, 0\}$
$u \leftarrow$ Deque(MRJ)
$t \leftarrow 1$
while MRJ not empty **do**
  $v \leftarrow$ Deque(MRJ)
  $d \leftarrow$ EuclideanDistance(u, v, t)
  if $d > R$ **then return** 0
  else
    $u \leftarrow v$
    $t \leftarrow t + 1$
return 1

We modify the simulation algorithm slightly to enable it to run hybrid EG-RAODV. Recall that the MRJ is a path along the cluster heads. Thus, the simulation algorithm must be tweaked to allow cluster members to pass the messages along to the next cluster until the message finally arrives at the destination node's CH, who subsequently sends the message to the destination node.

We implement our simulation and perform all data analysis in Python. We use a number of Python packages including snap, numpy, and scikit-learn.

A. *Results*

Table 1 shows the results of the EG-RAODV simulations on our preprocessed data set using transmission range $H = [300m, 400m, ..., 700m]$. The **avg time** row is the average number of seconds to compute the MRJ and the **avg hops** row shows the average number of hops in the MRJ. The **path found** row is the ratio of the simulations where the EG-RAODV algorithm was able to find a MRJ whose reliability score met our threshold. And the **success rate** row shows the rate of simulations where the message was successfully transmitted from the source node to the destination node.

As the results show, the EG-RAODV algorithm achieves high success rates for values of H >300m. The algorithm achieves higher success rates as we increase H but the computation time also increases. Note, however, that in this simulation, the rate of growth of the MRJ computation starts to slow. This is because, even though the average node degree increases quadratically with H, the MRJ path length also gets shorter. Eventually, if we continue to increase H, the computation time will start to decrease since the message can be transmitted in a just one or two hops. However, this is a consequence of the simplicity of the simulation, which fails to consider factors such as interference from all of the nodes transmitting at very high power and saturating the spectrum. Thus, the simulation is less realistic for larger values of H.

H	300m	400m	500m	600m	700m
avg time	0.38	1.51	2.36	3.40	3.61
avg hops	19.68	18.60	15.67	13.02	10.80
path found	1.0	1.0	1.0	1.0	1.0
success rate	0.65	0.72	0.82	0.90	0.99

Table 1 - EG-RAODV Results

Table 2 shows the results of our hybrid EG-RAODV approach. The column **h(600m)** refers to the heuristic-based clustering approach using a transmit range H = 600m. Column **h(1000m)** is the heuristic-based clustering using H=1000m.

algorithm	h(600m)	h(1000m)	spec
avg time	0.0296	.0157	0.014
avg hops	17.2	11.41	26.0
path found	1.0	0.96	0.342
success rate	0.707	0.985	0.17
num clusters	690	332	600
avg cluster size	4.37	9.08	5.03

Table 2 - Hybrid EG-RAODV Results

The table shows that the hybrid approach achieves a lower success rate of 0.707 at H=600m than the original EG-RAODV approach, which had a success rate of 0.9 for the same value of H. This can be explained by reasoning that the CH nodes essentially form a down-sampled version of the full network. Recall in Figure 6 that the network achieves a connectivity greater than 0.9 when H is approximately 500m. Down sampling the network effectively shifts the plot in Figure 6 to the right, which lowers the connectivity of the network and reduces the success rate of transmissions.

To test this theory, we raise H to 1000m and rerun the algorithm. As column h(1000m) shows, it achieves a success rate of 0.985. Thus, the hybrid EG-RAODV approach can achieve comparable success rate to the original EG-RAODV approach by simply raising the transmit power.

In addition, we test the hybrid EG-RAODV approach using a common "out-of-the-box" clustering algorithm. Specifically, we use the spectral clustering implementation found in the scikit-learn Python package. We choose k=600, where k is the desired number of clusters and a transmit range of H = 1000m. Column **spec** shows the results using the spectral clustering.

The results show that an "out-of-the-box" algorithm may not work well. Using spectral clustering, only 34% of the random simulations resulted in a MRJ that had reliability score that met our threshold, and of that 34%, only 17% of them succeeded. These results are not surprising since the spectral clustering does not take into account relevant factors such as transmit power. Additionally, the spectral clustering algorithm is based on forming minimum cut connections. Ultimately, we would want there to be many cuts between clusters, as that would increase the odds of a path existing between clusters, improving reliability.

All of the approaches hybrid EG-RAODV results show that the average MRJ computation time is orders of magnitude smaller than that of the original algorithm. In addition, it should be noted that the taxi data does not have particularly high node density. We expect the advantage of the hybrid EG-

RAODV approach to be much more apparent in a network with high density. This strongly suggests that the hybrid approach is much more practical in real VANET networks.

VIII. Conclusion

In this paper, we apply the EG-RAODV algorithm on data derived from a real data set of taxi cabs in Beijing. Our simulations results show that the algorithm may be an effective way to route messages in real VANETs, achieving nearly nearly 100% transmission success rate with a transmit range of 700m. Additionally, we propose a hybrid approach that combines popular VANET clustering techniques with the EG-RAODV algorithm. The hybrid approach achieves approximately 77% transmission success rate on with a 600m range and near 99% with a 100m while being orders of magnitude faster than the original EG-RAODV algorithm. This suggests that the hybrid approach is a viable alternative that requires less computation time as well as less network overhead.

Additionally, we investigated a clustering approach based on spectral clustering. However, spectral clustering yielded lower performance due to the fact that it doesn't take into consideration power requirements to form links in clusters and the fact that it attempts to minimize the cut between clusters, which can result in diminished reliability.

Furthermore, we would like to explore the relationship between the number of clusters and the success rate of the MRJ algorithm. Our clustering algorithm produces the minimum number of clusters for a given transmission radius. However, we would expect that by increasing the number of clusters we could retain the efficiency of clustering while improving its performance. As well, we would like to investigate other more sophisticated routing methods which would allow multi-hop routing within a cluster, so the clusters could take MRJ's to their respective cluster heads, rather than just trying one hop.

References

[1] M. H. Eiza and Q. Ni, "Evolving Graph Based Reliable Routing Scheme for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 52, pp. 1493–1504, May 2013.

[2] M. Sood and S. Kanwar, "Clustering in manet and vanet: A survey," in *International Conference on Circuits, Systems, Communications and Information Technology Applications*. CSCITA, 2014, pp. 375–380.

[3] c. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A comparative survey of vanet clustering techniques," in *IEEE Communications surveys And Tutorials*. IEEE, 2017, pp. 657–681.

[4] Y. Zheng, "T-drive trajectory data sample," August 2011. [Online]. Available: https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/

[5] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. New York, NY, USA: ACM, 2011, pp. 316–324. [Online]. Available: http://doi.acm.org/10.1145/2020408.2020462

[6] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: Driving directions based on taxi trajectories," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '10. New York, NY, USA: ACM, 2010, pp. 99–108. [Online]. Available: http://doi.acm.org/10.1145/1869790.1869807