A Clustering-Based Fast and Stable Routing Protocol for Vehicular Ad Hoc Networks

To cite this article: P F Zhao et al 2018 J. Phys.: Conf. Ser. 1060 012050

View the article online for updates and enhancements.

Related content
- A survey on different privacy-preserving authentication schemes in VANET
  Deepu Mathew and Hima Anns Roy
- UMDR: Multi-Path Routing Protocol for Underwater Ad Hoc Networks with Directional Antenna
  Jianmin Yang, Songzuo Liu, Qipei Liu et al.
- AD HOC Networks for the Autonomous Car
  Davidescu Ron and Eugen Negrus
A Clustering-Based Fast and Stable Routing Protocol for Vehicular Ad Hoc Networks

P F Zhao¹,²,³, K Liu¹,²,³, Y Zhang¹,²,³, T Zhang¹,²,³ and F Liu¹,²,³

¹ School of Electronics and Information Engineering, Beihang Univ., Beijing, China
² Beijing Key Lab for Network-Based Cooperative ATM, Beijing, China
³ Beijing Laboratory for General Aviation Technology, Beijing, China

zpf_wqt@163.com

Abstract. For the sake of improving network performances, i.e. end-to-end delay and packet drop ratio (PDR), in vehicular ad hoc networks (VANET), we propose a clustering-based fast and stable routing (CFSR) protocol. The CFSR protocol introduces a link quality assessment mechanism that evaluates the link quality on road segments between intersections and adopts the end-to-end delay as an evaluation metric to assign a weight to the link corresponding to each road segment. The delay-sensitive application can directly use this weight as the link quality indicator to directly establish the routing path in the sub-zone by using Dijkstra algorithm. At the same time, the concept of local coordinator (LC) is introduced into the routing protocol to help construct the routing path conveniently and quickly. In addition, it can also avoid the local optimum problem caused by single-step decision and reduce the network transmission delay. In this paper, the Simulation of Urban Mobility (SUMO) and Network Simulation 3 (NS3) are used to simulate this protocol. Simulation results reveal that CFSR protocol can obtain lower end-to-end delay and PDR than those of the AODV and DSDV protocols.

1. Introduction

Vehicular ad hoc network (VANET) basically is a specific form of mobile ad hoc network (MANET) applied in modern transportation system, serving as providing communication among the adjacent vehicles and the nearby fixed road-side equipment. VANET includes two communication approaches, Vehicles-to-Vehicles (V2V) and Vehicles-to-Infrastructure (V2I). Some applications of VANET focus on improve the driving safety via reminding drivers of avoiding dangerous transportation events, such as car crashes, extreme weather and so on, while others pay attention to making trips more joyful and relaxing. For instance, passengers can acquire real-time information about traffic statistics so as to select a faster route to save time. In addition, business services of malls and restaurants can be accessed through road side units (RSUs) [1].

There is a rising demand for a lower end-to-end delay (E2ED) for real-time non-safety applications. That is to say, it is crucial to choose a routing protocol with minimum E2ED to handle this issue. Most existing routing protocols select the routing paths by using greedy-based algorithms which are likely to account for local maximum problem, leading to higher E2ED [2][3]. For example, most position-based routing protocols, such as GPSR and GSR, select the shortest distance path between source and destination, while GyTAR and A-STAR choose roads which are well connected. As said before, these protocols that use greedy algorithm and single-step decision are prone to the local maximum problem...
which occurs when no other connected roads are closer to the destination than the current one. Although this issue can be settled down by carry-and-forward mechanism, it incurs longer delivery delay.

Giving preference to the global network topology is our motivation to design a clustering-based fast and stable routing (CFSR) protocol, which can avoid the local optimal problem. To achieve this goal, the CFSR protocol introduces a link quality assessment mechanism that evaluates the link quality on road between intersection and uses the end-to-end delay as an evaluation indicator to assign a weight to the link corresponding to each road, as designed in [4]. The delay-sensitive application can directly use this weight as the link quality indicator to directly establish the routing path in the sub-zone by using Dijkstra algorithm. In this way, we can achieve the goal of reducing the E2ED.

2. Network Model
This model aims at a typical city environment and assumes that it merely consists of road segments and intersections as shown in figure 1. For this network model, the article adopts a hierarchical structure to implement network division and management. Specifically, the road segment can be divided into several stationary clusters, meaning that the vehicles on the specific road segment belong to relative clusters [5]. Clusters can communicate with each other through cluster heads; and communication between vehicles of different road segments can be relayed through the gateways node selected at the intersections. Here are some assumptions and explanations for the typical scenario.

- The hierarchical cluster structure is shown in figure 2 and can be divided into three parts. The underlying network consists of cluster members (CMs) within each road segment. The middle-level network consists of the cluster heads (CHs) of each link and the gateways of the crossroads. The high-level network consists of local coordinators (LCs) selected by each sub-zone topology.
- $R_{ij}$ is defined as the road between intersections $i$ and $j$. At the same time, the hierarchical network can be mapped into a graph $G(V, E)$, where $V$ is a set of network nodes $v$ and $E$ is a set of edges $e$ between nodes. In addition, $V'$ is a path set, and if this set exists a path $e_i, e_a, e_b, … , e_n, e_j$ between the vehicle nodes $i$ and $j$, then $R_{ij}$ is called as “connected” state.
- Assume that nodes in the network are all assigned a state: cluster head (CH), cluster member (CM), gateway node (GW) or local coordinator (LC).
- Each node has a unique ID, equipped with GPS devices.

![Figure 1. Network model.](image1)

![Figure 2. Hierarchical clustering structure.](image2)

3. Clustering mechanism
The road segment is designed as a bi-direction road, and each is divided into multiple cluster. Moreover, the diameter of clusters equals half of the transmission range of a standard vehicle. As shown in figure 3, due to the particularity of the lateral extension of the road, the length of a cluster is much larger than its width, which results in that the cluster shape on the road can be approximated as a rectangle.
3.1. Clustering head selection process

After the stationary cluster division ends, the cluster head selection process begins. Vehicles within a cluster interact information \(<ID, x, y, v, d, b>\) - where ID is the identification of each vehicle, \(<x, y>\) is the European coordinates of the vehicle, \(v\) is the speed, \(d\) is the direction, and \(b\) is the vehicle node flag to indicate the status of the vehicle node (LC: 11, GW: 10, CH: 01, CM: 00). Based on these interactions, this paper selects two factors, velocity and location, as the cluster head selection criteria to calculate stability of nodes [6].

3.1.1. Velocity factor. This performance of speed is crucial in VANET. It results in high dynamics of the cluster structure and very fast change of the cluster topology. If the node with strong mobility is selected as the cluster head, it will cause the instability of the communication link and the cluster structure. The optimal result makes us choose a node as the vehicle node with smallest speed differences between neighbor vehicles.

Supposing that the CH selection mechanism takes the instantaneous speed \(v_i\) for each vehicle, where \(v_u\) is the average of all vehicle nodes at time \(t\) for calculating the speed weighting factor \(P_v(i)\) for each vehicle.

\[
P_v(i) = \frac{v(i) - v_u}{v_v}, \forall i \in \phi_n
\]  

For a vehicle node within a cluster, its velocity weighting factor can be expressed as a vector in the following formula:

\[
P_v(CID) = [P_v(1), P_v(2), P_v(3), ..., P_v(n)]
\]  

These values can be normalized by scaling them between zero and one as:

\[
P_{v_{norm}}(i) = \frac{P_v(i) - \min(P_v(CID))}{\max(P_v(CID)) - \min(P_v(CID))}, \forall i \in \phi_n
\]

The vehicle with the least \(P_{v_{norm}}(i)\) value has the highest speed priority.

3.1.2. Location factor. When the position of the cluster head is close to the start cluster boundary position, the cluster head will stay for a longer time to prevent frequent replacement of cluster heads, which can improve the stability of the cluster. Therefore, we also consider using position weight factor as one of metrics.

The location factor is defined as the distance from current position to stationary cluster boundary, the formula is as follows:

\[
P_l(i) = \frac{L(i)}{R}, \forall i \in \phi_n, 0 \leq L(i) \leq R
\]
For a vehicle node within a cluster, its location weight factor can be expressed as a vector in the following formula:
\[ P_l(CID) = [P_l(1), P_l(2), P_l(3), \ldots, P_l(n)] \]  
(5)

These values can be also normalized by scaling them between zero and one as:
\[ P_{norm}(i) = \frac{P_l(i) - \min(P_l(CID))}{\max(P_l(CID)) - \min(P_l(CID))}, \forall i \in \phi_n \]  
(6)

The vehicle with the least \( P_{norm} \) value has the highest location priority.

3.1.3. Integrated stability factor. For one cluster, the integrated stability of vehicles can be obtained by:
\[ S(i) = \alpha P_{norm}(i) + (1-\alpha)P_{norm}(i), \alpha \in (0,1) \]  
(7)

The former part of the formula evaluates the impact of distance between vehicles on cluster stability, because the greater the distance from the boundary of the vehicle, the more frequently the cluster heads are replaced, which is not conducive to cluster stability. The latter part mainly reflects the impact of vehicle speed on vehicle stability, because when the vehicle speed is small, the topology of the network changes slowly and the service lifetime is relatively long. On the other hand, if the vehicle speed is relatively fast, the performance of the network life will deteriorate. In general, the smaller the value of the stability \( S \), the more stable the cluster.

3.2. Construction of clusters
Assuming that vehicle nodes on the road are equipped with GPS, information of the vehicle such as speed, position, and direction can be obtained from the respective on-board GPS. Therefore, at any time, each vehicle node can confirm which cluster it belongs to, based on the location information. The ID of the vehicle is generated randomly and is not repeated. And we set the initial state of all nodes to the UNDIFIED state. Cluster construction flow chart shown in figure 4.

3.3. Intersection gateway selection
The division of the segmented road clusters and the CH selection are described above. At this time, the communication on the segmented roads is basically established. However, when the data transmission crosses a road segment to an intersection, the communication should be transmitted without interruption. Therefore, this paper selects gateway node at the intersection to perform relay transmission of information. Assuming that the vehicle nodes will obtain the turn direction (turn left, turn right, go straight) itself through the turn signal when each vehicle crosses the intersection, and use the reactive location service (RLS) to obtain its current location. Then, the node located in the intersection area will calculate its own residence time according to formula (8). The criteria for selecting the gateway node at the intersection is to make the vehicle node as long as possible to ensure the relay of information. For instance, the gateway node is more inclined to select the vehicle node as a gateway node that spends the longest time passing through the intersection. The selected gateway node will announce its own status and establish a connection with the neighboring cluster head.

As shown in figure 5, for the calculation of the dwell time of intersections at the intersection, the following are specifically divided into three categories: straight, left, and right.

\[ t_i = \begin{cases} 
\frac{W - d_{i,d}}{v_o}, & 0 \leq d_{i,d} \leq W, \text{ straight} \\
\frac{3W - 2d_{i,d}}{2v_o}, & 0 \leq d_{i,d} \leq \frac{3W}{2}, \text{ left} \\
\frac{W - 2d_{i,d}}{2v_o}, & 0 \leq d_{i,d} \leq \frac{W}{2}, \text{ right} 
\end{cases} \]  
(8)

Vehicle nodes determine whether they are located in the intersection area according to their own position information, if located at the intersection, calculate and interact with their respective dwell
time according to the current direction of rotation, the final selected gateway node requires the longest residence time. After the gateway node is selected successfully, it sends a HEAD frame to establish a connection with the cluster head nodes in the neighboring area.

**Figure 4.** Flow chart of cluster construction.

**Figure 5.** Intersection gateway selection.

### 4. CFSR protocol

Based on completion of the cluster structure, we start to establish the routing path for the data transmission. To achieve this goal, the process of assessing link quality and assigning weight to each link is necessary. Then, the weight can be directly used as the link quality metric to establish an optimal routing path in every sub-zones.

#### 4.1. Link quality assessment

The quality of each road link will be evaluated and the weight $w_R$ will be assigned according to the end-to-end delay of each road segment.
where $d_R$ is the transmission delay of a Link Assessment Packet (LAP), $d_e$ is the extra delay when a LAP encounters a disconnected path and caused by the carry-and-forward strategy, and $d_{rap}$ ($d_{rap} = t_r - Timestamp$) is the propagation delay which is related with the received time and timestamp of LAP. $T_s$ ($T_s = \sum_{i=1}^{\max} t_i$) is the sum of transmission delays of road segments along the path, and $T_{max}$ ($T_{max} = \frac{L}{v_{av}}$) is maximum tolerable transmission delay.

$d_R$ represents the average delay for transmitting a new data packet over a road segment. It is defined as below.

$$d_R = \sum_{i=1}^{n} d_{ci}$$  \hspace{1cm} (10)\]

$d_{ci}$ consists of two parts: $T_q$: representing the queuing delay and $T_{ts}$: denoting the transmission delay.

$$d_{ci} = E[t_q + t_{ts}]$$  \hspace{1cm} (11)

Actually, $T_q$ and $T_{ts}$ are correlated. The queuing delay of a packet corresponds to the service time of those packets queued ahead of it, which indicates that when a new packet arrives to a queue containing $k$ packets, $d_{ci}$ can be defined as below.

$$d_{ci} = (k + 1)E[t_{ts}]$$  \hspace{1cm} (12)

In addition, the extra delay $d_e$ can be interpreted as the product of additional delay and the disconnected probability.

$$d_e = d_{dis-i} \times \rho_{dis-i}$$  \hspace{1cm} (13)

$d_{dis-i}$ is the additional delay to incur when choosing carry-and-forward. To compute it, we divide $L$ into $n$ sub-segments of length $R$, meaning $n$ clusters as shown in figure 6. We assume that the cluster is connected if it contains at least one vehicle. Thus, the total additional delay of $m$ disconnected sub-segments is expressed as below.

$$d_{dis-i} = \frac{m \times R}{v_i}, m \leq n$$  \hspace{1cm} (14)

Figure 6. Extra delay caused by carry-and-forward.

$\rho_{dis-i}$ represents the probability of carry-and-forward. We use Poisson process as the model of the vehicle arrivals to road segments. Hence, we obtain:

$$\rho_{dis-i} = C_m^m \times \rho_{null-i}^{m} \times (1 - \rho_{null-i})^{m-n}$$  \hspace{1cm} (15)

$$\rho_{null-i} = \exp\left(\frac{\Delta}{\lambda}\right)$$  \hspace{1cm} (16)

Based on all above, we can assign weights to road segments as below.

\[
\begin{cases}
  d_R, & d_{rap} \leq T_s \\
  d_R + d_e, & T_s < d_{rap} < T_{max} \\
  \infty, & d_{rap} \geq T_{max}
\end{cases}
\hspace{1cm} (9)
\]
\[ w_{R_p} = \begin{cases} 
    d_{R_p}, & d_{R_p} \leq T_s \\
    d_{R_p} + d_{\text{div}} \times p_{\text{div}}, & T_s \leq d_{R_p} \leq T_{\text{max}} \\
    \infty, & d_{R_p} \geq T_{\text{max}} 
\end{cases} \]  

(17)

4.2. Local coordinator selection

The method to solve the path local optimum problem is to understand the global topology structure and weight information of the entire network, but it is extremely difficult given the considerable size of modern city. Therefore, we divide the urban network topology into sub-zones and establish a local coordinator in every one of them through information exchange. Considering this, we set the size of sub-zone to 3*3 blocks for the sake of limiting LAPs flooding and complexity of algorithm.

The local coordinator is defined as the specific gateway node that connects the most intersection gateway nodes in their respective scopes. In doing so, on the one hand, the efficiency of route discovery is high. On the other hand, it also avoids the problems of excessive information storms and maintenance overhead of the LAPs.
As shown in figure 7, in the initialization phase, each gateway node in the local-area topology sets itself as LC, and sets the number \( n \) of other gateway nodes connected to itself to zero. After the first round of LAP information interaction, each gateway node updates its neighboring link information and \( n \) value according to its own received LAP condition and compares it with other gateway nodes, mainly comparing the number of its connecting gateway nodes, \( n \) determines the coordinator after the first round of comparison. For example, in figure 7(a), the gateway nodes A, G, and H specify B, D, and I as their coordinators, and B, D and F designate E as their own coordinator. For A, B, they have the same value of \( n \), then select the node with the lower total weight as the coordinator. In the second round of interactions, each gateway node embeds its own routing table in the LAP (options field) before broadcasting. Such interaction process continues until the local area network generates a common coordinator. As shown in figure 7(c), the gateway node E can reach any gateway node within three hops, and therefore the gateway node E is selected as the local coordinator.

5. Performance evaluation

5.1. Simulation environment

This paper uses the SUMO [7] and NS3 [8] simulation platform to build a typical 3*3 urban road simulation scenario as shown in figure 8 and 9 to verify the performance of the CFSR protocol which compares with DSDV [9] and AODV [10]. It assumes that there is an intersection, two-direction carriageway, four lanes in each direction, including left-turn lanes, straight lanes, and right-turn lanes. The length of each road segment is 1000m, and a total of 9 crossroads are set.

| Parameters              | Value         |
|-------------------------|---------------|
| Numbers of vehicles     | 500           |
| Average speed of vehicles (km/h) | 40           |
| Communication range (m) | 50 ~ 350      |
| Simulation time (s)     | 300           |

The simulation parameters are set out in Table 1 above. In the table, the total number of vehicles in the 3*3 sub-zone network is 500, which is generated by different nodes. The average speed of the
urban vehicles is set to 40 km/h, assuming that the communication range of each vehicle is 50-350 m. The simulation time is set to 300s.

![Figure 8. Urban road simulation scenario.](image1)

![Figure 9. Urban road simulation scenario (SUMO).](image2)

5.2. Performance metrics

We propose two performance metrics to evaluate the CFSR protocol, including end-to-end delay \( (D_c) \) and packet drop ratio \( (p_d) \). \( D_c \) is defined as the average duration of a packet from source to destination. \( p_d \) is the ratio of discarded packets to total existing packets. The relevant formulas are as follows.

\[
D_c = \frac{1}{N_{success}} \sum_{i=1}^{N_{success}} D_i, \quad p_d = \frac{N_{failure}}{N_{failure} + N_{success}}. \tag{18}
\]

where \( N_{success} \) is the number of packets that are sent successfully, \( N_{failure} \) is the number of dropped packets, and \( D_i \) is the delay of the \( i \)th packet.

5.3. Simulation results

In the section, we compare CFSR protocol with AODV and DSDV, and simulation results are as follows.

![Figure 10. End-to-end delay.](image3)

![Figure 11. Packet dropping rate.](image4)

The change of E2ED relative to the communication range is shown in figure 10. We have observed that the average E2ED of all protocols increases as the communication range becomes larger. Since both AODV and DSDV use hop-by-hop forwarding, they can all lead to data congestion and thus increase E2ED. Since the CFSR uses a clustering structure, the data forwarding only passes through the cluster head and the gateway node, so the efficiency is higher and a lower E2ED is provided.

The relationship between the packet delivery rate and the maximum communication range of the node is shown in figure 11. We observe that the PDR for all protocols decreases as the range of communication. The PDR of AODV and DSDV is poor because the cost of route maintenance becomes large for large-scale networks and the high dynamics of the vehicle, the efficiency becomes very low, the link detection is not timely, and thus many packets are lost. The CFSR does not need to
maintain every node. It only needs to maintain the relevant information of the cluster head and
gateway nodes. In the route discovery phase, the CFSR can quickly query the local coordinator related
routes to establish the routing path. The efficiency is high, and the link is reliable, so PDR is higher.

6. Conclusion
In this paper, we propose the CFSR protocol based on the clustering structure, introduce the link
quality assessment, the selection process of the local coordinator, and establishes and updates the
route based on the weight distribution. In the sub-zone, we use Dijkstra algorithm to build the
routing path. On the one hand, the optimal path of the routing is selected in the case of ensuring
acceptable maintenance overhead, and on the other hand, sub-zone division is used to solve the local
optimum problem and maintenance overhead of LAPS proliferation. Finally, performance comparison
of the AODV, DSDV and CFSR protocols is made by using SUMO and NS3. Simulation results show
that CFSR has lower E2ED and higher PDR than the other protocols.

Acknowledgments
This work was supported in part by the National Key R&D Program of China under Grant No.
2016YFB1200100, NSFC under Grant Nos. 61231013, 61271195 and 61301083, National Science
Foundation for Innovative Research Groups of China under Grant No. 61521091.

References
[1] Zeadally S, Hunt R, Chen Y S, et al. Vehicular ad hoc networks (VANETS): status, results, and
challenges[J]. Telecommunication Systems, 2012, 50(4): 217-241
[2] Cheng J J, Cheng J L, Zhou M C, et al. Routing in Internet of Vehicles: A Review[J]. IEEE
Transactions on Intelligent Transportation Systems, 2015, 16(5): 2339-2352
[3] Sharef B T, Alsaqour R A, Ismail M. Vehicular communication ad hoc routing protocols: A
survey[J]. Journal of Network & Computer Applications, 2014, 40(1): 363-396
[4] Togou M A, Hafid A, Khoukhi L. SCRP: Stable CDS-Based Routing Protocol for Urban
Vehicular Ad Hoc Networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2016,
17(5): 1298-1307
[5] Abuashour A, Kadoch M. Performance Improvement of Cluster-Based Routing Protocol in
VANET[J]. IEEE Access, 2017: 15355-15371
[6] Nasr M M M, Wang Z G, Shen L F. VANET Clustering Based Routing Protocol Suitable for
Deserts[J]. Sensors, 2016, 16(4): 1-23
[7] SUMO: Simulation of Urban Mobility, [Online]. Available: https://www.dlr.de/ts/sumo/en/
[8] The Network Simulator: ns3. [Online]. Available: https://www.nsnam.org/
[9] Perkins C E, Bhagwat P. Highly dynamic Destination-sequenced Distance-Vector routing
(DSDV) for mobile computers[C]// Conference on Communications Architectures, Protocols
and Applications. ACM, 1994: 234-244
[10] Perkins, BeldingRoyer. Ad hoc On-Demand Distance Vector (AODV) Routing[J]. Internet-
Draft, draft-ietf-manet-aodv-13.txt, 2003, 6(7):90