TMC: Exploiting Trajectories for Multicast in Sparse Vehicular Networks

Ruobing Jiang, Yanmin Zhu*, *Member, IEEE,* Xin Wang, *Member, IEEE,* Lionel M. Ni, *Fellow, IEEE*

Abstract—Multicast is a crucial routine operation for vehicular networks, which underpins important functions such as message dissemination and group coordination. As vehicles may distribute over a vast area, the number of vehicles in a given region can be limited which results in sparse node distribution in part of the vehicular network. This poses several great challenges for efficient multicast, such as network disconnection, scarce communication opportunities and mobility uncertainty. Existing multicast schemes proposed for vehicular networks typically maintain a forwarding structure assuming the vehicles have a high density and move at low speed while these assumptions are often invalid in a practical vehicular network. As more and more vehicles are equipped with GPS enabled navigation systems, the trajectories of vehicles are becoming increasingly available. In this work, we propose an approach called *TMC* to exploit vehicle trajectories for efficient multicast in vehicular networks. The novelty of *TMC* includes a message forwarding metric that characterizes the capability of a vehicle to forward a given message to destination nodes, and a method of predicting the chance of inter-vehicle encounter between two vehicles based only on their trajectories without accurate timing information. *TMC* is designed to be a distributed approach. Vehicles make message forwarding decisions based on vehicle trajectories shared through inter-vehicle exchanges without the need of central information management. We have performed extensive simulations based on real vehicular GPS traces and compared our proposed *TMC* scheme with other existing approaches. The performance results demonstrate that our approach can achieve a delivery ratio close to that of the flooding-based approach while the cost is reduced by over 80%.

Index Terms—sparse vehicular networks, multicast, trajectory, encounter prediction.

1 Introduction

Recent advances in short-range radio technology such as Dedicated Short Range Communications (DSRC) [1] [2] for inter-vehicle communications have driven significant efforts in investigating and developing vehicular networks. By sharing information among moving vehicles, a vehicular network can support a wide variety of real-world applications, including emergence alert [3], advertisement, file sharing [4] [5], data collection [6], etc.

Information and message exchanges through multicast, where packets are sent from one sender to a group of receivers, have gained popular use and serve as a crucial routine operation in vehicular networks. For example, the taxis in a city may form an information network where each taxi may collect various types of information such as road surface condition, road closure status due to maintenance and traffic accidents. For more efficient information dissemination, a taxi can subscribe for some types of information of its interest, while a vehicle that collects the relevant information can serve as a source to disseminate the collected data through multicast to the group of subscribers.

Different from conventional communication networks, vehicular networks exhibit many unique characteristics, which pose several great challenges to efficient multicast. *First*, as vehicles may distribute over a vast area, the number of vehicles in a given region can be very limited. Therefore, a vehicular network can be sparse and resemble a delay tolerant network (DTN) [7] that relies on the "carry-and-forward" paradigm to exchange information among vehicles. In a sparse network, it is very difficult to find a connected path between any pair of vehicles. *Second*, as two vehicles can communicate only when they encounter (i.e., within the communication range of each other), the encounter opportunities become the critical network resources, which are usually scarce [8]. This makes it necessary for a multicast approach to be cost efficient. *Finally*, there is great uncertainty with vehicle mobilities, which makes it difficult to predict the future location of a given vehicle.

A few approaches [9] [10] [11] have been proposed for multicast in vehicular networks. However, stemming from multicast schemes proposed for Mobile Ad Hoc Networks (MANETs), these approaches often require maintaining a costly forwarding structure such as a tree or a mesh. They typically assume that vehicles in the network are densely populated and move at a lower speed. Both assumptions may be

• *Manuscript received on Sep 5, 2013, and accepted on Jan 1, 2014.*
• *Ruobing Jiang and Yanmin Zhu are with the Department of Computer Science and Engineering at Shanghai Jiao Tong University and Shanghai Key Lab of Scalable Computing and Systems. Xin Wang is with the Department of Electrical and Computer Engineering at Stony Brook University. Lionel M. Ni is with the Department of Computer Science and Engineering at Shanghai Jiao Tong University as well as Department of Computer Science and Engineering at Hong Kong University of Science and Technology.*
• **Yanmin Zhu is the corresponding author, and his email is yzhu@cs.sjtu.edu.cn.*

invalid in a practical vehicular network which often has sparse connections in an area, thus making these multicast approaches inefficient and even fail.

On the other hand, some potential opportunities have not been exploited for vehicular communications. Vehicles are increasingly deployed with Global Positioning System (GPS) enabled navigation systems. A recent report shows that approximately 300 million GPS devices have been shipped in 2009 alone [12]. The GPS enabled navigation system can suggest a path towards a destination. When the future trajectory of a vehicle is known in advance, its mobility uncertainty is greatly reduced. Two vehicles may potentially encounter each other if their trajectories intersect with each other. This observation suggests that the knowledge of trajectories of vehicles may be applied to predict future encounters, which will in turn bring valuable information to guide more efficient message forwarding in vehicular networks. More specifically, the forwarding may be more efficient if a message is forwarded to the set of vehicles that can potentially encounter more destination nodes, i.e., the relay vehicles would have a higher capability of delivering this message to destination nodes.

Motivated by this observation, we propose *trajectory-based multicast (TMC)* which exploits vehicle trajectories for more efficient multicast transmissions in sparse vehicular networks. We focus on information dissemination among public vehicles such as taxis and buses which run for the most time of a day. We have conducted empirical study based on real GPS traces from around 2,000 taxis in Shanghai, China, and find that there are on average 5.6 encounters between each pair of taxis in one day when the communication range is $200\,m$.

In *TMC*, a novel message forwarding metric is proposed to characterize the capability of a vehicle to forward a given message to a group of destination nodes, which is defined as a vector of delivery potential of the message to each of the destination nodes. With this metric, a vehicle can simply forward a message to a vehicle that has a higher multicast delivery gain over the vehicle itself. To compute the metric, the key challenge is to predict the chance of encounter between two vehicles based only on their trajectories without accurate timing information. To conquer this challenge, we model the travel time of a vehicle as a Gamma-distributed random variable and verify the modeling with real vehicular GPS traces. Then, a novel method is designed to predict the chance of inter-vehicle encounters. The salient feature of *TMC* is that it is a fully distributed approach in which vehicle trajectories are shared through inter-vehicle exchange and a vehicle makes its message forwarding decision based on the trajectories it learns instead of relying on a central point for information management.

To the best of our knowledge, this is the first work that exploits trajectory information to perform efficient multicast in sparse vehicular networks. The main technical contributions are as follows.

- We propose a novel message forwarding metric for multicast in sparse vehicular networks, which characterizes the capability for a vehicle to deliver the message to multiple destination nodes.
- We provide a method to predict the chance of inter-vehicle encounters based on only trajectory information by modeling the travel time of a vehicle as a Gamma-distributed random variable.
- We have performed extensive simulations based on real vehicular GPS traces, and compared our approach with other existing approaches. Our results have demonstrated that the proposed *TMC* approach can achieve a delivery ratio close to that of the flooding-based approach while the cost is reduced by more than 80%.

The remainder of the paper is organized as follows. Section 2 describes the design of our approach. We address the two key issues of our approach in Section 3 and Section 4, respectively. The performance evaluation is presented in Section 5. Section 6 reviews related work. We conclude the paper in Section 7.

2 DESIGN OF *TMC*

In this section we give the design details of *TMC*. First, we define the network model. Then, we present the basic idea of the design. Next, we introduce the message forwarding metrics and the message forwarding procedure. Finally, we highlight the key issues of *TMC*. Some other design issues are discussed in the online supplementary file.

2.1 Network Model

We consider a vehicular network consisting of N nodes. Two vehicles can communicate when they encounter each other, i.e., when they are within the communication range of each other. We do not require that the vehicles are densely populated, thus the network connectivity can be unavailable.

A message m to multicast has the following attributes: message ID, source node s_m, the set D_m of destination nodes, and the time-to-live (TTL) limit beyond which the message will be dropped.

As in [13] [14], the future trajectory T_i of a vehicle i is obtained from the GPS enabled navigation system when the driver inputs the destination for getting the driving path. We assume most of the drivers follow the driving paths suggested by the navigation system. Trajectory T_i of vehicle i consists of a sequence of road segments and is associated with a starting point. Trajectory T_i of each vehicle i and the starting point are disseminated in the vehicular network by trajectory sharing when vehicles encounter each other.

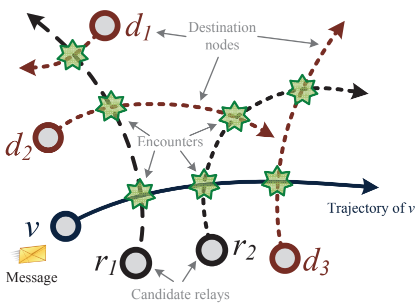


Figure 1. Illustration of the basic idea of *TMC*.

2.2 Basic Idea

The basic idea of *TMC* is to forward a message to vehicles with a higher capability of delivering the message to more destination nodes. In another word, a relay node of a message should be able to encounter more destination nodes.

As an example, in Fig. 1, the trajectories of vehicles v, r_1, r_2, d_1, d_2 and d_3 are shown. At the intersections of different trajectories, encounters occur between the two corresponding vehicles. Vehicle v has a message to forward to a destination set including vehicle d_1, d_2 and d_3, and it will encounter vehicle r_1, r_2 and d_3. After encountering v, vehicle r_1 will encounter d_1 and d_2, and vehicle r_2 will encounter d_2 and d_3. Vehicle v is able to forward the message to the destination node d_3 when v and d_3 encounter each other. As vehicle r_1 can forward the message to both d_1 and d_2, v should forward the message to r_1. In contrast, v will not forward the message to r_2 in spite that r_2 will encounter d_2 and d_3 because d_2 and d_3 have already been taken care jointly by v and r_1.

2.3 Message Forwarding Metrics

For the success of *TMC*, it is critical to design an efficient *message forwarding metric* which characterizes the capability of a vehicle to deliver a given message to the set of destination nodes. With the metric, a vehicle v can compute the *multicast delivery gain* for a candidate relay node r based on which v can determine if it should forward the message to r.

We next give the formal definitions of the two message forwarding metrics, the *delivery potential vector* of a candidate relay node r to a message m and the *multicast delivery gain* of r over vehicle v.

Definition 1 (Delivery Potential Vector)**.** *Given a message m, the delivery potential vector $\vec{\gamma_m}(v)$ of a vehicle v is defined as the vector of delivery probabilities that vehicle v delivers m to the destination nodes in D_m:*

$$\vec{\gamma_m}(v) \triangleq \langle p_1(v), p_2(v), \cdots, p_k(v) \rangle, \tag{1}$$

where $k = |D_m|$, the ith position of the vector represents the ith destination node in D_m sorted in the increasing order of their IDs, and $p_i(v)$ is the corresponding delivery probability for the ith destination node.

Note that $p_i(v)$ is zero when v has no potential of delivering message m to destination node i.

Definition 2 (Multicast Delivery Gain)**.** *Given a message m, the multicast delivery gain of vehicle r over vehicle v, denoted as $\phi_m(r, v)$, is defined as,*

$$\begin{aligned} &\phi_m(r, v) \triangleq \sum_{i=1}^{k} \alpha_i, \\ &\textit{where } \alpha_i = \begin{cases} p_i(r) - p_i(v), & \textit{if } p_i(r) - p_i(v) > 0 \\ 0, & \textit{otherwise} \end{cases}, \end{aligned} \tag{2}$$

From the definition, we can see that the multicast delivery gain of r over v reflects the additional benefit for delivering m to destination nodes if r is selected as a relay. It is reasonable to ignore the negative effect of r on delivering m to destination i when $p_i(r) - p_i(v) \leq 0$. The justification is as follows. In our multicast routing algorithm, message replication is used, which indicates that a message remains in the forwarder vehicle and a copy is replicated on the relay vehicle. With this in mind, we actually want to select those relays which can take the message to any one of the destination nodes with high probability. If a potential relay has a high delivery probability of relaying the message to one of the destinations while has low delivery probabilities for the rest destinations, this relay is still desirable since it contributes to the success of the multicast of the message. On the other hand, if we only forward messages to those relays that have high delivery probabilities for all destinations, then the number of such relays would be very small. As a result, the overall delivery probability of the message would be very small.

2.4 Message Forwarding Procedure

When the message forwarding metrics are available, we are ready to describe the trajectory-based multicast routing. That is, how does the vehicular network forward messages to their respective destination vehicles. Essentially, the core of the multicast routing process is the procedure executed by each vehicle once it encounters another. Through period beacons at the MAC layer, a vehicle can discover the presence of a new neighbor and the departure of an existing neighbor. Once discovering a new neighbor, a vehicle executes the procedure in which the vehicle decides whether to forward its messages to the new neighbor and the order of these messages being forwarded.

Each vehicle gives forwarding priority to those messages with higher delivery gains with respect to the new neighbor. To explain the procedure, without loss of generality, we consider vehicle v encounters another vehicle u. Upon discovering the new neighbor u, node v should decide whether to forward its messages to u, and the order of these messages being forwarded. Consider a message denoted by m in vehicle v. To

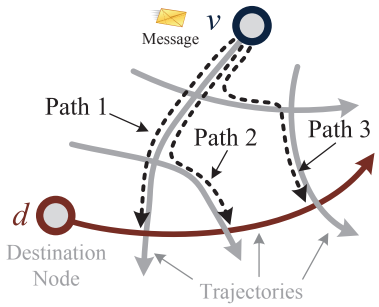


Figure 2. Illustration of the computation of the delivery probability of v delivering m to a given destination d.

enable u to compute the delivery gain for m, it maintains a vector $\vec{\gamma_m}$ of maximum delivery probabilities that have been recorded in previous relays to each destination. The computation of delivery gain requires encounter probabilities which needs the knowledge of vehicle trajectories. Such knowledge is obtained by each vehicle by sharing trajectories and corresponding starting points each time two vehicles encounter. By v forwarding $\vec{\gamma_m}$ to u, u can compute its gain based on the trajectories it has maintained. Only when u has a positive delivery gain over $\vec{\gamma_m}$, v will forward m to u.

Considering v might have multiple messages to forward to u while the encounter duration might not be long enough for all the messages to be transmitted, v should decide the message forwarding order. The messages are sorted according to their delivery gains. The pseudo codes of the message forwarding procedure for both vehicle v and u can be found in the online supplementary file.

It is possible that in the collision domain of v, there might be other vehicles contending for the channel in addition to u. This paper does not focus on the design of MAC protocols. Note that the adopted MAC protocols for inter-vehicle communication should include periodic beacon mechanism for neighbor discovery.

2.5 Key Issues

To implement *TMC*, it is necessary for each vehicle v to compute its delivery potential vector $\vec{\gamma_m}(v)$ for a given message m. Essentially, v should compute its delivery probability of delivering m to each destination node d, $p_d(v)$. To compute $p_d(v)$, the following two key issues must be addressed.

- **Computation of delivery probability**. The delivery probability $p_d(v)$ of vehicle v delivering m to destination node d is dependent on all the future inter-vehicle encounters in the network. Section 3 addresses this issue.
- **Prediction of inter-vehicle encounters**. We need to compute the encounter probability of two vehicles when their trajectories intersect with each other. In spite of the knowledge of future trajectories, one does not know the accurate arrival time of the vehicle at a specific location on the trajectory, which is critical for estimating the encounter probability of two vehicles. This issue is addressed in Section 4.

3 Computing Delivery Potential Vector

To derive $\vec{\gamma_m}(v)$, there is a need to calculate the delivery probability $p_d(v)$ of delivering m to each destination d. In this section, we provide the detailed procedures of deriving $p_d(v)$.

We first illustrate the main idea of computing $p_d(v)$ in Fig. 2, where trajectories of several vehicles including v and d are shown. There are three possible forwarding pathes $Path\,1, Path\,2$ and $Path\,3$ for the message m carried by v to reach d. Whether the message m can be delivered along a path is probabilistic because the encounter at the intersection of two trajectories is uncertain. Let λ_i denote the $Path\,i$ and $P(\lambda_i)$ denote the delivery probability along the forwarding path i. Then $p_d(v)$ can be computed as,

$$p_d(v) = 1 - \prod_{i=1,2,3} (1 - P(\lambda_i)) \,. \tag{3}$$

As a result, to compute $p_d(v)$ we should first obtain the following two items.

- Item 1: The set of all possible forwarding paths connecting v and the destination node d, denoted by $\Theta_v(d)$.
- Item 2: The message delivery probability along each forwarding path $\lambda_i \in \Theta_v(d)$, i.e., $P(\lambda_i)$.

We propose the trajectory-based encounter graph to derive the two items in the following.

3.1 Trajectory-based Encounter Graph

We denote the trajectory of a vehicle i by T_i, containing the geographic information of the driving path of i and associating with a starting time.

Definition 3 (Trajectory-based Encounter Graph)**.** *A trajectory-based encounter graph $G = \{V, E_u, E_b\}$ is constructed based on a set Ψ of vehicular trajectories, including a vertex set V, a unidirectional edge set E_u and a bidirectional edge set E_b. Along a trajectory $T_i \in \Psi$, for each intersection with another trajectory $T_j, j \neq i$, there is a vertex $\rho_j^i \in V$ which is associated with a random variable of vehicle i's arrival time at the intersection. Between two successive vertices ρ_j^i and $\rho_k^i, k \neq i$ along T_i, there is a unidirectional edge $\vec{e} \in E_u$ from ρ_j^i to ρ_k^i. Between any pair of vertices ρ_j^i along T_i and ρ_i^j along T_j there is a bidirectional edge $e \in E_b$ which indicates that the two vehicles i and j can potentially meet and exchange messages at the intersection .*

A vehicle v constructs a trajectory-based encounter graph $G(v)$ based on the trajectory set $\Psi(v)$ maintained by v. $\Psi(v)$ is updated when v encounters

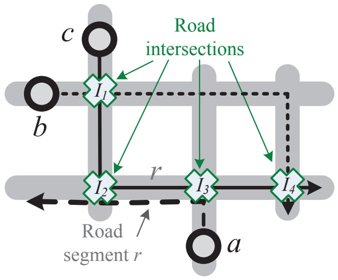


Figure 3. Trajectories of vehicles a, b, and c in a small road network.

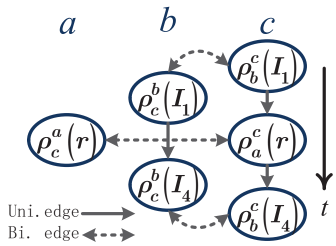


Figure 4. The trajectory-based encounter graph based on the three trajectories shown in Fig. 3.

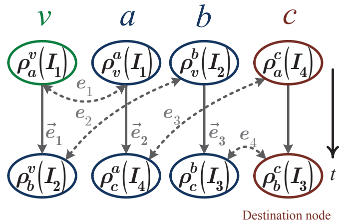


Figure 5. An example trajectory-based encounter graph to illustrate the main process of the searching algorithm.

other vehicles which share the trajectory information with v. To create the set $V(v)$ of $G(v)$, every two trajectories $T_i, T_j \in \Psi(v)$ are compared for locating their intersections. Each intersection results in a pair of vertices ρ_j^i and ρ_i^j.

Each vertex $\rho_j^i \in V(v)$ is associated with a random variable τ_j^i of i's arrival time at the intersection of T_i and T_j. The probability distribution of the random variable will be discussed in the subsection 4.1. It is possible that the same pair of trajectories have more than one intersections, so we attach spatial and temporal information to each vertex for differentiation.

A unidirectional edge $\vec{e} \in E_u(v)$ between ρ_j^i and ρ_k^i indicates that a vehicle i first meets the vehicle j and then meets the vehicle k on its trajectory T_i. As the message is carried by the vehicle i during its traveling between the two successive intersections, the probability of moving the message between the two intersections is 1. Thus the weight of $\vec{e}$, $P(\vec{e})$, is set to 1. A bidirectional edge $e \in E_b(v)$ between ρ_j^i and ρ_i^j indicates that there is an chance for the vehicles i and j to encounter and exchange messages. The weight of e, $P(e)$, represents the encounter probability and is less than 1. The way of estimating the encounter probability will be introduced in Section 4.

As an example, we show the trajectories of three vehicles a, b and c in a small road network in Fig. 3. The corresponding trajectory-based encounter graph G is shown in Fig. 4. Along the trajectory T_c of vehicle c, there are two intersections with the trajectory T_b at road intersections I_1 and I_4 respectively, and an intersection with the trajectory T_a on the road segment r. Thus, there are three corresponding vertices $\rho_b^c(I_1), \rho_b^c(I_4)$ and $\rho_a^c(r)$ in G. We differentiate the two intersections between the trajectories of c and b, i.e., $\rho_b^c(I_1)$ and $\rho_b^c(I_4)$, with the geographic positions I_1 and I_4. Along T_c there are three successive intersections represented by vertices $\rho_b^c(I_1), \rho_a^c(r)$ and $\rho_b^c(I_4)$ respectively. Accordingly, there are two unidirectional edges in G, one is from $\rho_b^c(I_1)$ to $\rho_a^c(r)$ and the other is from $\rho_a^c(r)$ to $\rho_b^c(I_4)$. There are also two successive intersections along T_b at I_1 and I_4, respectively. Correspondingly, there is a unidirectional edge from $\rho_c^b(I_1)$ to $\rho_c^b(I_4)$. For the trajectory intersection between T_a and T_c on the road segment r, there is a bidirectional edge between vertices $\rho_c^a(r)$ and $\rho_a^c(r)$.

3.2 Searching for Forwarding Paths

With $G(v)$, an efficient searching algorithm is applied to find all possible paths from v to each destination.

We next explain the main process of the searching algorithm with an example trajectory-based encounter graph shown in Fig. 5. The searching algorithm introduces a parameter of *search depth* $\omega \geq 1$, which is defined as the maximum number of forwarding hops of all resulting paths. A forwarding hop corresponds to the message transmission between two vehicles, and the number of hops on a forwarding path reflects the number of times the messages are forwarded from one vehicle to another over the whole path.

The main process of the searching algorithm on $G(v)$ given a searching depth ω is as follows.

- *Initialization*: The starting point ρ_0 of the search is the first trajectory intersection along T_v, i.e., the first vertex of v. In our example, the starting vertex is $\rho_a^v(I_1)$.
- *Depth-1 search*: The vehicle v looks for the destinations that it can directly reach without need of another vehicle to forward the message. Thus, starting from ρ_0, v only needs to search along its unidirectional edges to find the vehicles that can be encountered in one hop, denoted by β_1. Once encountering these vehicles, v can forward the message to them. In the example, $\beta_1 = \{a, b\}$ because $\rho_a^v(I_1)$ and $\rho_b^v(I_2)$ are found.
- *Depth-i search* ($1 < i \leq \omega$): The search goes along the bidirectional edges from vehicles in β_{i-1} to find the vehicles that can be reached within i hops from ρ_0. From a vehicle in β_{i-1}, same as the *Depth-1 search*, the search goes along unidirectional edges to find vehicles that have not been included by β_{i-1} to create the set β_i of vehicles to encounter in i hops. In the sample, $\beta_2 = \{c\}$ and $\forall i > 2, \beta_i = \emptyset$.
- *Termination*: After the search finishes, the destination nodes and the forwarding path rooted at

v are available. In our example, there are two forwarding pathes from v to the destination node c, i.e., $\rho_a^v(I_1) \rightarrow e_1 \rightarrow \rho_v^a(I_1) \rightarrow \vec{e}_2 \rightarrow \rho_c^a(I_4) \rightarrow e_3 \rightarrow \rho_a^c(I_4)$ and $\rho_a^v(I_1) \rightarrow \vec{e}_1 \rightarrow \rho_b^v(I_2) \rightarrow e_2 \rightarrow \rho_v^b(I_2) \rightarrow \vec{e}_3 \rightarrow \rho_c^b(I_3) \rightarrow e_4 \rightarrow \rho_b^c(I_3)$.

According to the searching algorithm, the search depth ω controls the tradeoff between the estimation accuracy of the message delivery probability and the computation complexity. When a larger search depth is used, more destination nodes and more forwarding paths from a vehicle to each of the destination nodes would be included. However, a larger search depth leads to a higher computation complexity. We know that the message delivery probability through a forwarding path quickly decreases with the increasing number of hops. As a result, a small search depth suffices in practice, e.g., 3 hops.

3.3 Calculating $P(\lambda_i)$

Suppose $\lambda_i \in \Theta_v(d)$ is a forwarding path searched by the previous algorithm and λ_i contains a set of bidirectional edges, denoted by $\Phi(\lambda_i)$, then $P(\lambda_i)$ can be computed as follows,

$$P(\lambda_i) = \prod_{e \in \Phi(\lambda_i)} P(e). \tag{4}$$

It is possible that when two vehicles, e.g., v and u, encounter with each other, neither v nor u can find a forwarding path to deliver a message of v to one of the destinations, e.g., i. In another word, both $p_i(v)$ and $p_i(u)$ are zero. Then, the multicast delivery gain of forwarding the message from v to u is computed based only on those destinations to which u has higher delivery probabilities than v. As a result, the destination i, to which both v and u have a zero delivery probability, will be ignored.

4 Predicting Inter-vehicle Encounters

The occurrence of an encounter between two vehicles requires two conditions. First, there is a trajectory intersection between the two trajectories. Second, the arrival instants of the two vehicles at the intersection position are so close that the two vehicles will reside within the communication range of each other. Based on the two conditions, we compute the encounter probability of two vehicles given their trajectories.

4.1 Modeling of Travel Time

When predicting inter-vehicle encounters, it is necessary to have the knowledge of a vehicle's arrival time at an intersection on its trajectory. However, vehicular trajectories do not provide such arrival times. Fortunately, the vehicular travel time over a route of urban roads follows the Gamma distribution, as suggested in some prior research [15] [16].

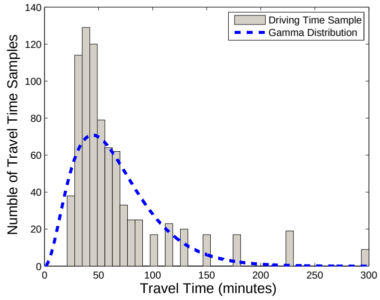


Figure 6. Travel time distribution of a randomly selected road segment.

The probability density function (PDF) of the Gamma distribution can be expressed in terms of the gamma function parameterized in terms of a shape parameter κ and a scale parameter θ. The equation defining the PDF of a gamma-distributed random variable τ, representing the vehicular travel time, is

$$f(\tau;\kappa,\theta) = \frac{1}{\theta^{\kappa}} \frac{1}{\Gamma(\kappa)} \tau^{\kappa-1} e^{-\frac{\tau}{\theta}}, \quad \tau \geq 0, \text{and } \kappa,\theta > 0. \tag{5}$$

To verify that the vehicular travel time follows Gamma distribution, we have conducted empirical study based on real vehicular GPS traces from around 2,058 taxis in Shanghai, China. For a randomly selected road segment, we compute the travel time of about 1,000 vehicles which forms a sample set. Based on the sample set, we first plot a histogram in Fig. 6 which shows the distribution of all the samples. Then we estimate the two parameters of the PDF of the Gamma distribution by conducting the maximum likelihood estimation over the sample set. With the estimated parameters, we plot the PDF of this specific Gamma distribution in Fig. 6, too. The Kolmogorov-Smirnov (K-S) test shows that the sample set follows Gamma distribution with a significance level of 95%.

As a result, we model the travel time over a path as a random variable following the Gamma distribution in which there are two parameters. We derive the two parameters We estimate the two parameters by conducting statistics on the real vehicular GPS traces as follows. First, travel time samples for the driving path are collected, which include both the time on road segments along the path and the time at corresponding intersections. Then, maximum likelihood estimation is conducted based on the travel time samples to achieve the parameters.

4.2 Calculation of Encounter Probabilities

The rationale behind the computation of encounter probabilities is as follows. Suppose the trajectories of vehicle v and u intersect at position I. The arrival time of v at I, denoted by $\tau_u^v(I)$, has been introduced in the construction of encounter graphs (Section 3.1). $\tau_u^v(I)$ can be modeled as a Gamma distributed random variable. This is because the starting time instant of v's trajectory is known and the vehicular travel time until v arrives at I can be modeled as a Gamma distributed random variable as introduced in the previous subsection. Similarly, the arrival time of u at I, $\tau_v^u(I)$, is also a random variable. We denote the time difference between $\tau_u^v(I)$ and $\tau_v^u(I)$ by $\Delta(\tau_u^v(I), \tau_v^u(I))$. To ensure that u and v can encounter at position I, $\Delta(\tau_u^v(I), \tau_v^u(I))$ has an upper bound, denoted by δ. When $\Delta(\tau_u^v(I), \tau_v^u(I))$ is larger than δ, u and v cannot encounter because they will be out of the communication range of each other when they arrive at I. δ can be determined based on the transmission range and the relative moving speed of two vehicles.

We consider two types of inter-vehicle encounters. We next explain the computation of the encounter probability for each type. The first type of encounters occur at a road intersection, e.g., the encounter between b and c at I_1 in Fig. 3. Denoting the PDF of a random variable τ by f_τ, the encounter probability can be computed as,

$$\begin{aligned} & Pr\{b \text{ encounters } c \text{ at } I_1\} \\ = \; & Pr\{\Delta(\tau_c^b(I_1), \tau_b^c(I_1)) < \delta\} & (6) \\ = \; & \int_0^\infty \int_{t-\delta}^{t+\delta} f_{\tau_c^b(I_1)}(t) \times f_{\tau_b^c(I_1)}(t')dt'dt. & (7) \end{aligned}$$

Eq. 6 means the probability that b and c encounter at I_1 is the probability that the time difference between the arrival time of b and c at I_1 is shorter than δ.

The second type of encounters take place when two vehicles move in different directions on the same road, e.g., the encounter between vehicles a and c on road r in Fig. 3. Denoting the arrival time of a at road intersections I_2 and I_3 by $\tau^a(I_2)$ and $\tau^a(I_3)$, respectively, and those of c at I_2 and I_3 as $\tau^c(I_2)$ and $\tau^c(I_3)$, respectively, we have

$$\begin{aligned} & Pr\{a \text{ encounters } c \text{ on road } r\} \\ = \; & Pr\{\tau^a(I_3) < \tau^c(I_3) \cap \tau^c(I_2) < \tau^a(I_2)\} & (8) \\ = \; & \int_0^\infty \int_t^\infty f_{\tau^a(I_3)}(t) \times f_{\tau^c(I_3)}(t')dt'dt \\ & \times \int_0^\infty \int_t^\infty f_{\tau^c(I_2)}(t) \times f_{\tau^a(I_2)}(t')dt'dt. & (9) \end{aligned}$$

This means that vehicles a and c encounter each other when both the time a entering r ($\tau^a(I_3)$) is earlier than the time c leaving r ($\tau_c(I_3)$) and the time c entering r ($\tau_c(I_2)$) is earlier than the time a leaving r ($\tau_a(I_2)$).

5 Performance Evaluation

We evaluate the performance of *TMC* in this section. More evaluation results can be found in the online supplementary file.

5.1 Methodology and Experimental Setup

We have conducted extensive simulations based on real vehicular GPS traces, and compared our approach with three state-of-the-art related approaches. The GPS traces are collected from 2,058 taxis in Shanghai, China during a period of 32 days, covering the urban area of 130 km in length and 69 km in width.

To evaluate the routing performance, we use three metrics: *delivery ratio*, *transmission overhead*, and *delivery delay*. The metric of transmission overhead is an average over all the multicast sessions. For a single multicast session, the transmission overhead is the number of all the transmissions of data packets over the number of reached destination vehicles. The delivery delay is the average value of all the successfully delivered packets. The default value of system parameters in all the simulations are shown in Table 1.

Each multicast session has a source node generating a message with the TTL of two hours. The set of destination nodes are randomly selected and recorded in message header.

5.2 Compared Algorithms

- *Epidemic* [17]. The source node floods the message throughout the network to reach all multicast destination nodes.
- *RAPID* [18]. RAPID is originally designed for unicast in delay-tolerant networks. With RAPID, a message is forwarded to a relay who has a shorter expected delay to the destination. We have revised RAPID where a relay node is selected if it has a shorter delay to any of the destination nodes. Message replication is adopted.
- *STDF* [19]. This algorithm is initially designed for unicast in vehicular networks. It assumes the availability of real-time trajectories of all the vehicles. Encounters of two vehicles traveling on the same road segment and along opposite directions are considered. For fair comparison, STDF is also extended to support efficient multicast.

5.3 Overhead of Trajectory Sharing

In *TMC*, when two vehicles encounter each other, each of the vehicles will share with the other vehicle the information of currently maintained trajectories that the other vehicle does not have. As a result, the trajectory sharing process incurs transmission overhead. To learn how much the actual transmission overhead for sharing trajectories is incurred throughout the

TABLE 1
Default Settings of System Parameters

Parameter	Default Value
# of vehicles	1,000
# of multicast groups	50
# of destination nodes	20
Communication range	100 meters
Search depth ω	3

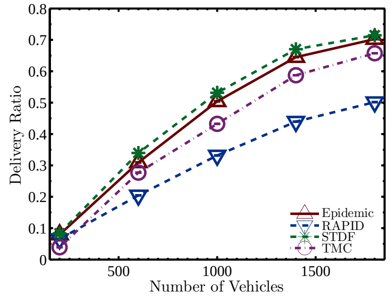


Figure 7. Delivery Ratio vs. Number of Vehicles.

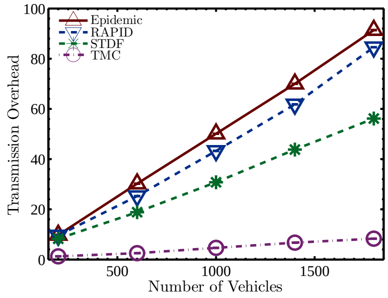


Figure 8. Transmission Overhead vs. Number of Vehicles.

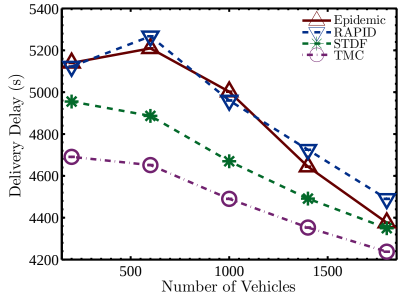


Figure 9. Delivery Delay vs. Number of Vehicles.

whole process, we have conducted trace driven simulations to study the overhead. We emphasize that the transmission overhead studied in this subsection only counts the amount of transmitted data for exchanging trajectories each time two vehicles encounter.

The experimental simulations are conducted based on the real vehicular GPS traces. We simulated the vehicular network for 3 hours (8:00am-11:00am). The amount of transmitted data on each encounter for exchanging trajectories was recorded each time two vehicles encounter each other. The number of vehicles is varied from 200 to 1,800. The communication range of vehicles is 100 meters. We use 140 bytes to store one vehicle trajectory because of the following analysis of the real traces. First, we can use four bytes to represent a road segment because there are less than 35,000 road segments in Shanghai, the largest city in China. Second, a trajectory of 10 hours contains about 34.2 road segments. As a result, 140 bytes are sufficient to represent a vehicle trajectory used in *TMC*.

Fig. 10 plots both the total transmission overhead of trajectory sharing of the whole network and the average overhead per vehicle in log scale as the number of vehicles in the network is varied. From the figure, we can find that the overhead of trajectory sharing is as low as hundreds of kilobytes per vehicle and increases with the number of vehicles. Therefore, the transmission overhead caused by sharing trajectories in *TMC* is very low. The main reason is that when

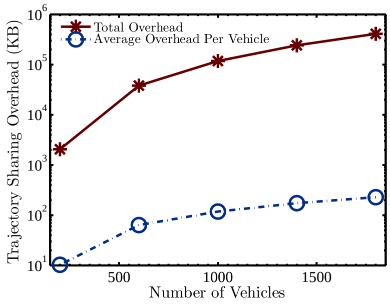


Figure 10. Trajectory sharing overhead vs. number of vehicles.

two vehicles encounter, only the trajectories that the other vehicle does not have are actually exchanged. Thus, it is practical to share trajectories for efficient multicast routing in vehicular networks.

In the rest of performance evaluation, the reported transmission overhead values all include the overhead caused by sharing trajectories among vehicles.

5.4 Impact of Number of Vehicles

We evaluate the performance of various schemes as the number of vehicles is varied from 200 to 1,800. Fig. 7, Fig. 8, and Fig. 9 plot the evaluation results.

In Fig. 7, we can see that *TMC* has a delivery ratio close to those of Epidemic and STDF and better than that of RAPID. For example, when there are 1,800 vehicles, the delivery ratio of *TMC* is only 6.4% and 8.0% lower than that of Epidemic and STDF, respectively. The delivery ratio of each scheme increases as the number of vehicles becomes larger because more nodes lead to a higher delivery capability of the network. STDF performs best because of the adoption of real-time vehicular trajectories. With the knowledge of real-time trajectories, inter-vehicle encounters are better estimated and and contribute to high delivery ratio. On the other hand, Epidemic floods data packets blindly. Thus, transmission chances cannot be efficiently used to achieve high delivery ratio. For *TMC*, the reason that the delivery ratio is lower than STDF is that the trajectories are shared through inter-vehicle transmissions. Thus, trajectories are not completely available to all the vehicles. For RAPID, it performs worst because it estimates the expected delay of relays based on less efficient statistic results of encounter history instead of real-time information.

As expected, in Fig. 8, *TMC* has the lowest transmission overhead among the compared algorithms. As *TMC* estimates not only the encounters between two vehicles on the same road segment, but also around road intersections which is ignored by STDF, *TMC* can find better sequences of relays with higher encounter probabilities. As a result, *TMC* deliver packets much more efficiently than STDF. When there are 1,800 vehicles in the network, the transmission

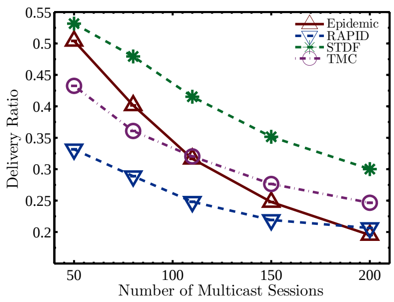


Figure 11. Delivery Ratio vs. Number of Multicast Sessions.

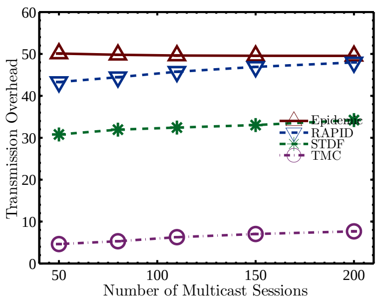


Figure 12. Transmission Overhead vs. Number of Multicast Sessions.

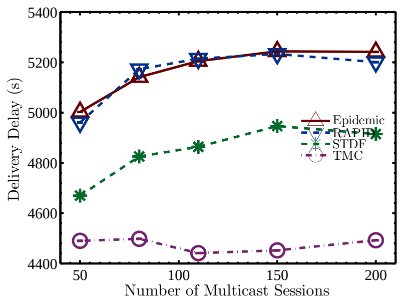


Figure 13. Delivery Delay vs. Number of Multicast Sessions.

overhead of *TMC* is 85.2%, 90.2% and 90.9% lower than that of STDF, RAPID and Epidemic, respectively. We can also find that trajectory based routing algorithms perform better than algorithms not using trajectories. This is because, with the knowledge of trajectories, packet transmissions to those vehicles with less probabilities to delivery the packet are reduced and precious encounter chances can be more efficiently used.

Fig. 9 shows that the delivery delay of four algorithms decrease as the number of vehicles decreases and *TMC* has the shortest delivery delay among all the compared algorithms. As mentioned above, the average delivery delay is computed over all the successfully delivery packets. In other words, for a multicast packet, the delivery delay for the packet to reach each received destination should be taken into computation. We can find that the trajectory based algorithms, i.e., *TMC* and STDF, have shorter delay than other algorithms. This is also because that, with the knowledge of trajectories, relay sequences with higher delivery probabilities are selected which avoids blind transmissions. When there are 600 vehicles, the delivery delay of *TMC* and STDF are 10.7% and 6.2% shorter than that of Epidemic, respectively.

5.5 Impact of Number of Multicast Sessions

In order to illustrate the sparse property with precious transmission chances in vehicular networks and to emphasize the importance of high efficient routing algorithms, we explore the impact of traffic load by varying the number of concurrent multicast sessions. Fig. 11, Fig. 12 and Fig. 13 show the results.

In Fig. 11, the delivery ratio of all the algorithms decrease as more concurrent multicast sessions exist. The most important feature in the figure is that the delivery ratio of Epidemic decreases dramatically with heavier traffic load. When there are 200 concurrent multicast sessions, Epidemic has the lowest delivery ratio which is 35% lower than that of STDF. As Epidemic blindly floods packets in the network without efficiently using the precious transmission chances, traffic loads higher than the transmission capacity of the network necessarily result low delivery ratio.

In Fig. 12, algorithms keep stable transmission overhead to successfully deliver a packet under different traffic loads. From the figure, we can find that trajectory based algorithms are more efficient than Epidemic and RAPID. For example, when there are 150 multicast sessions, the transmission overhead of *TMC* is 85.9% lower than that of Epidemic. We can also find than *TMC* is much more efficient than STDF. There are two reasons. First, *TMC* considers more encounter types than STDF. Thus, *TMC* can select better relays with higher delivery probabilities. Second, *TMC* is designed for multicast. *TMC* evaluates the delivery ability of a candidate relay based on its integrated delivery probabilities for all the destinations of a multicast packet. While STDF selects a relay only if it has a higher delivery probability for any destination.

As shown in Fig. 13, the average delivery delay of *TMC* for successfully delivery packets is the shortest. When the traffic load increases, the delivery delays of the three compared algorithms only increase slightly. This is because only a small ratio of packets are successfully delivered under heavy traffic load.

6 Related Work

We review related work in this section. Additional discussion on other related work including multicast algorithms in mobile ad hoc networks and previous studies in vehicular ad hoc networks are provided in the online supplementary file.

6.1 Multicast in Vehicular Networks

A few algorithms have been designed for multicast routing in vehicular networks, but most of them are extended from typical multicast routing protocols for MANETs, which makes them inappropriate for sparse vehicular networks.

ROVER [9] is a tree-based geographical multicast routing algorithm for vehicular networks. Vehicles in the zone of forwarding (ZOF) discover a forwarding tree reaching all the nodes in the zone of relevance (ZOR) rooted at the source node. It is clear that a high density of nodes is necessary for this protocol.

MDDV [11] is also a dissemination tree based algorithm. For each forwarding path to a specific destination region, a dynamically maintained group of nodes which locate closest to the destination region are forwarders.

6.2 Trajectory-based Routing in Vehicular Networks

Vehicular trajectories have been exploited for packet delivering in vehicular networks [20] since the availability of future trajectories significantly reduces the uncertainty with vehicular mobility.

TBD [14] is a routing approach for using trajectories to forward data from vehicles to a given roadside access point (AP) in a light traffic vehicular network. Each node estimates the delivery delay to the AP based on its trajectory, which is then used as the metric for making forwarding decisions. Wu et al. [21] propose to predict the future location of a vehicle by modelling the mobility of a vehicle as a multi-order Markov chain, and then estimate the encounter probability of each pair of vehicles. TSF [22] makes use of road side units (RSUs) and trajectories to forward data from a fixed roadside unit (RSU) to a moving vehicle.

In our work we also exploit vehicular trajectories for data delivery, but consider a different problem, i.e., multicast routing in sparse vehicular networks. Thus, our work is complementary to existing approaches.

7 Conclusion

In this paper we have presented an approach called *TMC* for efficient multicast in sparse vehicular networks. *TMC* employs a forwarding metric which characterizes the capability for a candidate relay node to deliver a message to each destination node in the multicast group. We predict the pairwise encounters of a vehicle with other vehicles to evaluate its delivery probability. *TMC* is a fully distributed approach, with which vehicles share their trajectories as they encounter each other. This makes *TMC* appealing for practical use in real vehicular networks. Our performance results demonstrate that *TMC* can achieve the packet delivery ratio close to that of a broadcast scheme with a much lower transmission overhead. We introduce our future work in the online supplementary file.

Acknowledgements

This research is supported by China 973 Program (2014CB340303), NSFC (No. 61170238, 60903190, 61027009, 60933011, 61202375), Doctoral Fund of Ministry of Education of China (20100073120021), National 863 Program (2009AA012201, 2011AA010500 and 2013AA01A601), HP IRP (CW267311), SJTU SMC Project (201120), STCSM (08dz1501600, 12ZR1414900), Shanghai Chen Guang Program (10CG11), Singapore NRF (CREATE E2S2), and MSRA funding for Urban Computing and for Star Track program. This work is also supported by Program for Changjiang Scholars and Innovative Research Team in University(IRT1158, PCSIRT), China.

References

[1] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 ghz dsrc-based vehicular safety communication," *IEEE Wireless Communications*, vol. 13, no. 5, 2006.

[2] *Dedicated Short Range Communications (DSRC) home*, http://www.leearmstrong.com/dsrc/dsrchomeset.htm.

[3] F. Farnoud and S. Valaee, "Reliable broadcast of safety messages in vehicular ad hoc networks," in *Proc. IEEE INFOCOM*, 2009.

[4] T. Kitani, T. Shinkawa, N. Shibata, K. Yasumoto, M. Ito, and T. Higashino, "Efficient vanet-based traffic information sharing using buses on regular routes," in *Proc. IEEE VTC Spring*, 2008.

[5] D. Zhang and C. Yeo, "Enabling efficient wifi-based vehicular content distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, 2013.

[6] A. Miloslavov and M. Veeraraghavan, "Sensor data fusion algorithms for vehicular cyber-physical systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, 2012.

[7] K. Fall and S. Farrell, "DTN: an architectural retrospective," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, 2008.

[8] Y. Ding, C. Wang, and L. Xiao, "A static-node assisted adaptive routing protocol in vehicular networks," in *Proc. ACM VANET*, 2007.

[9] M. Kihl, M. Sichitiu, T. Ekeroth, and M. Rozenberg, "Reliable geographical multicast routing in vehicular ad-hoc networks," *Wired/Wireless Internet Communications*, 2007.

[10] W. Kim and M. Gerla, "Cognitive multicast with partially overlapped channels in vehicular ad hoc networks," *Ad Hoc Networks*, 2012.

[11] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: a mobility-centric data dissemination algorithm for vehicular networks," in *Proc. ACM VANET*, 2004.

[12] H. Yomogita, *Mobile GPS Accelerates Chip Development*, http://techon.nikkeibp.co.jp/article/HONSHI/20070424/131605/.

[13] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "TBD: Trajectory-based data forwarding for light-traffic vehicular networks," in *Proc. IEEE ICDCS*, 2009.

[14] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. Du, "Trajectory-based data forwarding for light-traffic vehicular ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, 2011.

[15] R. Russell and T. Urban, "Vehicle routing with soft time windows and erlang travel times," *Journal of the Operational Research Society*, vol. 59, no. 9, 2007.

[16] I. Kaparias, M. Bell, and H. Belzner, "A new measure of travel time reliability for in-vehicle navigation systems," *Journal of Intelligent Transportation Systems*, vol. 12, no. 4, 2008.

[17] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proc. MobiSys workshop on Mobile opportunistic networking*, 2007.

[18] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *Proc. ACM SIGCOMM*, vol. 37, no. 4, 2007.

[19] F. Xu, S. Guo, J. Jeong, Y. Gu, Q. Cao, M. Liu, and T. He, "Utilizing shared vehicle trajectories for data forwarding in vehicular networks," in *Proc. IEEE INFOCOM*, 2011.

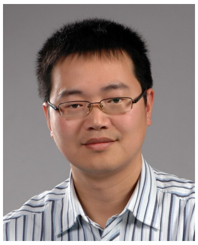
[20] D. Niculescu and B. Nath, "Trajectory based forwarding and its applications," in *Proc. ACM MobiCom*, 2003.

[21] Y. Wu, Y. Zhu, and B. Li, "Trajectory improves data delivery in vehicular networks," in *Proc. IEEE INFOCOM*, 2011.

[22] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "TSF: Trajectory-based statistical forwarding for infrastructure-to-vehicle data delivery in vehicular networks," in *Proc. IEEE ICDCS*, 2010.

[23] E. Royer and C. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *Proc. ACM MobiCom*, 1999.
[24] S. Sethi and S. Udgata, "SRMAODV: a scalable reliable maodv for manet," in *Proc. International Conference and Workshop on Emerging Trends in Technology*, 2010.
[25] S. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Mobile Networks and Applications*, vol. 7, no. 6, 2002.
[26] R. Biradar, S. Manvi, and M. Reddy, "Link stability based multicast routing scheme in manet," *Computer Networks*, vol. 54, no. 7, 2010.
[27] K. Viswanath, K. Obraczka, and G. Tsudik, "Exploring mesh and tree-based multicast routing protocols for manets," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, 2006.
[28] H. Xu and D. Mu, "A cluster based stable multicast routing protocol in ad hoc network," in *Proc. IEEE/IFIP EUC*, vol. 2, 2008.
[29] L. Law, S. Krishnamurthy, and M. Faloutsos, "A novel adaptive protocol for lightweight efficient multicasting in ad hoc networks," *Computer Networks*, vol. 51, no. 3, 2007.
[30] L. Ji and M. Corson, "Differential destination multicast-a manet multicast routing protocol for small groups," in *Proc. IEEE INFOCOM*, vol. 2, 2001.
[31] C. Gui and P. Mohapatra, "Scalable multicasting in mobile ad hoc networks," in *Proc. IEEE INFOCOM*, vol. 3, 2004.
[32] X. Xiang, X. Wang, and Y. Yang, "Stateless multicasting in mobile ad hoc networks," *IEEE Transactions on Computers*, vol. 59, no. 8, 2010.
[33] H. Zhu, M. Li, L. Fu, G. Xue, Y. Zhu, and L. M. Ni, "Impact of traffic influxes: Revealing exponential intercontact time in urban vanets," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, 2011.
[34] N. Lu, T. H. Luan, M. Wang, X. Shen, and F. Bai, "Capacity and delay analysis for social-proximity urban vehicular networks," in *Proc. IEEE INFOCOM*, 2012.
[35] T. V. Nguyen, F. Baccelli, K. Zhu, S. Subramanian, and X. Wu, "A performance analysis of csma based broadcast protocol in vanets," in *Proc. IEEE INFOCOM*, 2013.
[36] F. Martelli, M. Elena Renda, G. Resta, and P. Santi, "A measurement-based study of beaconing performance in ieee 802.11 p vehicular networks," in *Proc. IEEE INFOCOM*, 2012.
[37] Y. Wu, Y. Zhu, H. Zhu, and B. Li, "CCR: Capacity-constrained replication for data delivery in vehicular networks," in *Proc. IEEE INFOCOM*, 2013.
[38] X. Li, C. Qiao, X. Yu, A. Wagh, R. Sudhaakar, and S. Addepalli, "Toward effective service scheduling for human drivers in vehicular cyber-physical systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, 2012.
[39] S. Chang, Y. Qi, H. Zhu, J. Zhao, and X. Shen, "Footprint: Detecting sybil attacks in urban vehicular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, 2012.
[40] A. Gkoulalas-Divanis, P. Kalnis, and V. Verykios, "Providing k-anonymity in location based services," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, 2010.
[41] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J. Hubaux, "Secure vehicular communication systems: design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, 2008.

Ruobing Jiang is a PhD student in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. She received her BEng. degree from the Department of Computer Science and Engineering at Southeast University in 2010. Her research interests include vehicular ad hoc networks and mobile computing. She is a student member of the IEEE.



Yanmin Zhu is an Associate Professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. He obtained his PhD from the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology in 2007. Prior to joining Shanghai Jiao Tong University, he worked as a Research Associate with the Department of Computing at the Imperial College London, UK. His research interests include vehicular networks, wireless sensor networks and mobile computing. He is a member of the IEEE and the IEEE Communication Society.

Xin Wang received the B.S. and M.S. degrees in telecommunications engineering and wireless communications engineering respectively from Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree in electrical and computer engineering from Columbia University, New York, NY. She is currently an Associate Professor in the Department of Electrical and Computer Engineering of the State University of New York at Stony Brook, Stony Brook, NY. Before joining Stony Brook, she was a Member of Technical Staff in the area of mobile and wireless networking at Bell Labs Research, Lucent Technologies, New Jersey, and an Assistant Professor in the Department of Computer Science and Engineering of the State University of New York at Buffalo, Buffalo, NY. Her research interests include algorithm and protocol design in wireless networks and communications, mobile and distributed computing, as well as networked sensing and detection. She has served in executive committee and technical committee of numerous conferences and funding review panels, and is the referee for many technical journals. Dr. Wang achieved the NSF career award in 2005, and ONR challenge award in 2010.

Lionel M. Ni is the chair professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). He also serves as the special assistant to the president of HKUST, dean of the HKUST Fok Ying Tung Graduate School, and visiting chair professor of the Shanghai Key Lab of Scalable Computing and Systems at Shanghai Jiao Tong University. He has chaired more than 30 professional conferences and has received six awards for authoring outstanding papers. He is a fellow of the IEEE.

Appendix A
Pseudo Codes of the Message Forwarding Procedure

The pseudo codes of the message forwarding procedure are shown in Algorithm 1 and Algorithm 2. The procedure in Algorithm 1 is executed by every vehicle each time it discovers a new neighbor u. The procedure in Algorithm 2 is triggered by the discovering vehicle. The procedure includes the process for sharing trajectories between node v and u. The departure of the neighbor prematurely terminates the execution of the procedure.

Algorithm 1: Message Forwarding Procedure Executed on v

Notations M_i: the set of messages of vehicle i
Ψ_i: the set of trajectories maintained by vehicle i
I: maximum number of retransmissions
for iter=1, iter< I **do**
  $\forall m \in M_v$, broadcast ID$_m$, D_m, and $\vec{\gamma_m}$
  Share trajectories Ψ_v with u
  if Receive delivery gains from u **then**
    Decide the message forwarding order
    Broadcast messages in order
    $\forall m \in M_v$, update $\vec{\gamma_m}$,
    return
  end if
end for

Algorithm 2: Corresponding Procedure Executed on u

Notations M_i: the set of messages of vehicle i
Ψ_i: the set of trajectories maintained by vehicle i
if Receive message information from v **then**
  Share trajectories Ψ_u with v
  Compute delivery gains to messages involved in the received message information
  Broadcast delivery gains
  if Receive messages from v **then**
    Update M_u and $\vec{\gamma_m}, \forall m \in M_u$
  end if
end if

Appendix B
Additional Performance Evaluation on the Impact of Communication Range

In this set of simulations, we vary the communication range from 50 m to 250 m. Fig. 14, Fig. 15, and Fig. 16 plot the delivery ratio, transmission overhead and delivery delay, respectively.

As expected, in Fig. 14, the delivery ratio of each algorithm increases when larger communication range is adopted. This is because a larger communication range results more inter-vehicle contacts and a larger network capacity of packets delivery. Same as previous simulations, STDF has the best performance and *TMC* has a delivery ratio close to those of Epidemic and STDF. When the communication range is 150 m, the delivery ratio of *TMC* is only 12.2% lower than that of Epidemic, and 27.5% higher than that of RAPID. We can also find that STDF has similar delivery ratio as Epidemic. However, they use different schemes to achieve the best delivery ratio among all the approaches. Epidemic floods the messages over the network, taking full advantage of inter-vehicle encounters. Thus, Epidemic achieves high delivery ratio but with high transmission overhead, which will be shown later. While STDF relays messages in a more efficient way by assuming that each vehicle has the access to all the vehicular trajectories. Thus, STDF can largely reduce the transmission overhead.

In Fig. 15, Epidemic and RAPID keep stable transmission overhead under different communication ranges while trajectory based algorithms have slightly higher transmission overhead when the communication range increases. For example, the transmission overhead of *TMC* when the communication range is 150 m is 20.6% higher than that when the communication range is 100 m. The reason that trajectory based algorithms have higher transmission overhead under larger communication range is as follows. When larger communication range is adopted, more potential relays are available each time when a packet holder encounters with other vehicles. Before the best relay sequence is found, vehicles with higher delivery probabilities are selected as relays to improve the delivery ratio. Thus, more unnecessary transmissions are generated when larger communication range is adopted.

In Fig. 16, we can find that a larger communication range results in a shorter delivery delay for each algorithm. And when the communication range is large enough, the delivery delay of all the compared algorithms are close. When the communication range is 250 m, the delivery delay of Epidemic is only 3.7% and 6.7% higher than that of STDF and *TMC*, respectively. This is because a larger communication range leads to more inter-vehicle contacts and a larger network capacity of data delivery. Thus, the efficiency of algorithms becomes less important.

Appendix C
Multicast in MANETs

Multicast routing in mobile ad hoc networks (MANETs) has been extensively studied. The majority of multicast protocols can be classified into four categories, i.e., tree-based, mesh-based, cluster-based and stateless multicast routing.

A tree-based protocol constructs a forwarding tree [23] from the source to all multicast receivers. The routing performance will be significantly affected when network nodes are highly dynamic which results in frequent route failures. At the same time, a large number of control messages are generated to construct and maintain the forwarding tree, leading to a high overhead. MAODV [23] and its successor SRMAODV [24] are examples of tree-based protocol.

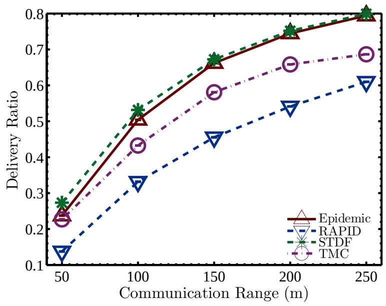


Figure 14. Delivery Ratio vs. Communication Range.

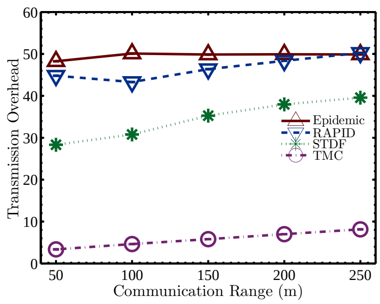


Figure 15. Transmission Overhead vs. Communication Range.

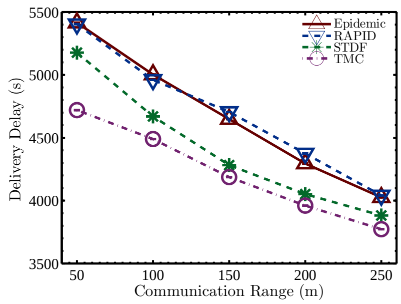


Figure 16. Delivery Delay vs. Communication Range.

Instead of constructing a tree which has a single path to each destination node, a mesh-based multicast protocol [25] [26] has more than one paths between the source node and each multicast receiver. The mesh topology thus can better cope with link failure [27], and mesh-based routing has a higher delivery ratio and lower delivery delay. However, redundancy forwarding paths bring heavy traffic load in a sparse network with constrained communication chance.

A cluster-based multicast protocol [28] builds local clusters or cohorts [29] of connected nodes to efficiently make multicast routing decisions.

Different from the previous categories, the stateless category of multicast protocols [30] [31] [32] does not rely on routing structures, thus minimizing the overhead for constructing and maintaining these routing structures. This category of protocols are especially suitable for highly dynamic networks.

Appendix D
Vehicular Networks

Many research efforts have been devoted to different aspects of vehicular networks. In [33], vehicle mobility in urban environments has been studied by looking at the inter-contact time which is reported to be exponentially distributed. Lu et al. [34] analyze the asymptotic capacity and delay performance of social-proximity urban vehicular networks with inhomogeneous vehicle density.

Some work focus on the link layer performance of vehicular networks. In [35], the performance of a CSMA based broadcast protocol in vehicular networks has been analyzed. And in [36], Martelli et al. study the beaconing performance of IEEE 802.110 based vehicular networks by analyzing the real measurements collected from field tests.

Many approaches for data delivery in vehicular networks have been proposed. In [5], a cooperative content distribution system called CCDSV is presented, which is based on a network of infrastructure APs to collaboratively distribute contents to vehicles. Zhu et al. find that there is temporal dependency for inter-vehicle contacts between vehicles and then design a prediction method to predict the next contact for a given pair of vehicles. with predicted contacts, vehicles can make better forwarding decisions. In [37], the optimal replication strategy has been studied for packet delivery in vehicular networks. In [6], sensor data fusion algorithms are developed to aggregate data collected vehicles.

In [38], Li et al. consider service scheduling problems in which a vehicle may not be able to obtain more than one service within a short period of time, and scheduling algorithms have been proposed. And urban vehicular networks are vulnerable to Sybil attacks and attack detection techniques are reported in [39].

Appendix E
Discussion

In the real world, vehicles may change their future trajectories. The possible reasons are traffic jams or accidents on their previous trajectories. When a vehicle changes its trajectory, the outdated ones that have been distributed in the network should be updated. Otherwise, the estimated delivery probabilities of a vehicle based on the information of the outdated trajectory would be inaccurate.

However, the impact of trajectory changes on the performance of our approach is limited. The main reasons are as follows. *First*, the outdated trajectories can be replaced quickly through our trajectory sharing mechanism. When a vehicle has a new trajectory, it shares its new trajectory with vehicles it will encounter. Gradually, the new trajectory will spread over the whole network to replace the outdated ones. We have conducted trace-driven simulations on the spread speed of trajectories sharing based on our real vehicular traces. The setting of the simulations follows the default values, as specified in our main file. The simulation results show that it only takes half an hour to spread 70% of all the trajectories to more than 73.2% of the vehicles in the network. *Second*, according to the computation of the message delivery probability (as described in the main file), this probability of a vehicle is determined by all the forwarding paths. If only one forwarding path is changed, the message delivery probability is only slightly changed. Finally, the fraction of vehicles which change their trajectories is usually small.

Appendix F
Future Work

We will carry our future work by investigating the privacy issue that may arise in our approach. The trajectory of a vehicle may disclose some sensitive privacy information of the vehicle driver. The current design does not consider the privacy issue in sharing vehicular trajectories. We will study privacy protection for individual trajectory in future work. Possible solutions include anonymization of vehicle trajectories [40] and restricted sharing among authorized vehicles [41].