FleaNet: A Virtual Market Place on Vehicular Networks (Invited Paper)

Uichin Lee, Joon-Sang Park, Eyal Amir, Mario Gerla
Department of Computer Science
University of California Los Angeles, CA 90095
{uclee,jspark,eyalamir,gerla}@cs.ucla.edu

***Abstract*— Over recent years, mobile Internet devices such as laptops, PDAs, smart phones etc, have become extremely popular and widespread. Once on board of a vehicle, these devices can automatically connect to the vehicle processor and thus greatly amplify the communications and processing capabilities available to the owner in a "pedestrian mode." We envision that this "amplification" opportunity will be one of the drivers of car to car and car to curb communications. In fact, the car communications system will not be used exclusively for mobile Internet access, but also as a distributed platform for the "opportunistic" cooperation among people with shared interests/goals. Exchanging safety messages among vehicles is a compelling example. Stretching opportunistic cooperation well beyond safety messages, we discuss in this paper the concept of virtual "flea market" over VANET called FleaNet. In FleaNet, customers, either mobile (i.e., vehicles) or stationary (i.e., pedestrians, roadside shop owner), express their demands/offers, e.g., want to buy or sell an item, via radio queries. These queries are opportunistically disseminated exploiting in part the mobility of other customers in order to find the customer/vendor with matching needs/resources. In the paper we identify the key performance metrics, namely query resolution latency, scalability, and mobility. Based on the metrics, using models and simulation, we show that FleaNet can efficiently support a market place over vehicular networks.**

I. Introduction

Wireless mobile devices such as smart phones, PDAs, and laptops become ubiquitous in our daily lives. We use them at home, while we walk and while we drive. In the future every vehicle will be equipped with wireless devices [2] that enable communications with roadside objects and also with other vehicles. These devices guide us into a new era of pervasive computing in which seamless access to information sources is provided. When traveling or shopping, for instance, we can search the *web* to get directions or to locate specific products. In fact, not only do such devices empower us with ubiquitous Internet access but they also create a new environment where opportunistic cooperation can emerge among users with shared interests/goals, e.g., drivers exchanging safety related information, shoppers/sellers trading goods, etc [11]. Using wireless vehicular devices, drivers will be able to cooperate to an extent and with a flexibility never known before.

This research described in this paper is related to an emerging body of work that exploits vehicular wireless connectivity. Previous works include PeopleNet [8] which enables wireless users to form a virtual social network that mimics the way people seek information via social networking through direct contacts. In PeopleNet, devices communicate with each other on behalf of the owners to exchange or obtain needed information. For example, an individual that wants to buy/sell a sold-out Yankees/Red Sox baseball ticket can simply "post" a buy/sell ticket query (for a stated price) at some location possibly far away from the Yankees' stadium. To permit the exchange of information over relatively large geographic areas, PeopleNet uses the fixed infrastructure (e.g., cellular networks, mesh networks, etc.) to post a query to its geographically pertinent place called a bazaar, e.g., the Yankees' stadium. Within a bazaar, it takes advantage of *free-of-charge* but intermittent connectivity provided by short range radios such as Bluetooth. This results in epidemic query dissemination. Namely, a query is propagated by simply copying it from one device to another whenever a wireless connection can be "opportunistically" established. If a match to the query is found, the user who initially placed the query is eventually informed of the match (e.g., via email or SMS message).

In this paper, we propose FleaNet to provide a "virtual flea market" service running in urban vehicular networks. FleaNet operates on the vehicular "ad hoc grid" without any infrastructure support. We will show that FleaNet provides an excellent method for people to communicate with each other as buyers and sellers of goods (or information) and to efficiently find matches of interest, potentially leading to transactions.

Urban vehicular networks formed by vehicles on the road and roadside stations in metropolitan areas can be characterized as large scale (e.g., the whole metropolitan area), dense (e.g., up to hundreds of thousands of nodes in a 100 square mile area), and highly mobile (e.g., up to 60 mph). These characteristics pose a formidable challenge to FleaNet. Epidemic query dissemination such as used in PeopleNet becomes less efficient because large amount of information might be *flooded* into the entire network due to very frequent car encounters in a large-scale vehicular network. In addition, while epidemically spreading, a given query could find the same matched query more than once, thus leading to potentially many redundant matches.

FleaNet remedies this problem with mobility assisted query dissemination where the query "originator" periodically advertises his query only to one hop neighbors. Each neighbor then stores the advertisement (i.e., query) in its local database without any further relaying; thus, the query spreads only because of vehicle motion. Upon receiving a query, a node tries to resolve it locally in its database; in case of success, the originator will be automatically informed. A match only happens in its neighbors and thus, there is no redundant match notification. This match could lead to an actual transaction; FleaNet also provides a mechanism that routes the transaction request/reply by using Last Encounter Routing (LER) [4].[1] LER is based on geo-routing and combines location service and routing service. In FleaNet the query packet includes the originator geo-coordinates, and thus, LER does not incur any additional routing cost.

The paper contributes new concepts/results to the existing body of research in this area. First, we introduce the new model of a virtual market place over vehicular networks. Second, we propose a novel epidemic based architecture, FleaNet that scales to thousands of nodes and is non-intrusive to existing services. Finally, we provide an extensive evaluation of the FleaNet protocols via both analysis and simulation. A key result of this study is the fact that a random query can be resolved, in most cases, within a tolerable amount of time and with minimal bandwidth, storage and processing overhead. Another interesting result is that if the advertiser, i.e., Adstation, is stationary, the query resolution time is critically dependent on its location.

This paper is organized as follows. In Section II, we review the related work. In Section III, we detail our proposed FleaNet followed by a simple analysis in Section IV. In Section V, we evaluate our protocols through extensive simulations. Finally, we conclude the paper in Section VI.

II. Related Work

A. Query dissemination

Mobile-to-mobile information exchange with infrastructure support has been addressed in PeopleNet [8]. PeopleNet mimics the way people disseminate and discover information in real life via personal contacts. It forms a virtual wireless social network through portable devices with multiple network interfaces, e.g., cellular and Bluetooth. In PeopleNet, a given area is divided into non-overlapping regions called bazaars each of which is dedicated to handling handle certain types of queries placed by users. A query (either sell or buy) is propagated via the network infrastructure, typically the cellular network, to k randomly selected users in the associated bazaar as the initial seeds for data dissemination. To spread queries in a sparse network in a given bazaar[2], PeopleNet uses epidemic query dissemination; a node randomly swaps a query with one of its neighbors because of rare encounters among people and relatively small size of buffers.

As mentioned earlier, PeopleNet is less efficient when it operates in a dense/large-scale network such as a VANET because the dissemination policy becomes network-wide flooding. One might argue that this problem can be alleviated by introducing probabilistic forwarding and a large advertisement interval (e.g., a received query could be disseminated with low probability), but such a policy is intrinsically not scalable as we will see later in Section IV. In addition, this epidemic dissemination will lead to potentially many "redundant" matches. In the worst case, given that a sell query has already disseminated to the whole users, say N, once a buy query starts spreading from the originator, it will generate $N-1$ redundant matches. These redundant matches will be notified to the originator and severely affect the performance. To these reasons, PeopleNet or similar schemes such as [13] cannot be directly applicable to vehicular networks.[3]

Instead, we utilize the two-hop *multi-copy* protocol, a mobility assisted data dissemination technique [3], where the source node only copies a message to its direct contact nodes (i.e., nodes within its communication range) and those node may forward it to the destination node. It is important to note that such a protocol is "scalable," as we will see later in Section IV. The key difference from the *multi-copy* protocol is that FleaNet does not assume any particular destination for a given message, but the message is disseminated over the network and intermidiate nodes with a matched query of shared interests are the potential destination of the message. Note that [3] also shows that the mobility pattern is crucial to the performance of data dissemination. Although restricted mobility in vehicular networks severely affects the performance, the popularity of a query can greatly improve the performance.

In AdTorrent [9], static wireless Digital Billboards on the roadside are used for advertisement. Digital Billboards only push ad contents to the vehicles passing by. Ad contents are potentially large in size such as hotel virtual tours or show previews. Each node gossips its content availability to neighbors to facilitate the search. On the other hand, mobile users search for contents of interest by querying neighbors up to k-hops away. As a result, they find potential nodes to download from (i.e., pull) the ad content. Since the encounters of Digital Billboards are rare, and the size of ad content is relatively large, ad consumers mostly "pull" the ad contents. Unlike AdTorrent, FleaNet is focused on disseminating and resolving small size queries (i.e., at most few KB), through which users satiate their market demands such as buying or selling an item. Since there is the equilibrium between supply and demand in a typical market, we could say that FleaNet is basically a "balanced" push/pull system.

B. Routing in VANET

In VANET, georouting has been extensively investigated for scalable delivery. Georouting works well in dense networks. However, if vehicle density is low, say off rush hour and in peripheral areas, the vehicle connectivity is intermittent. In this case, one can exploit the predictable mobility in VANET to "assist" georouting with *carry and forward*: a vehicle

[1] A user could see multiple matches for a given query. Based on his own criterion (either on distance from his current location or on the offered price), he selects the best one and sends the transaction request. Then, the target user responds with the transaction reply after seeing the request.

[2] This is mainly due to low penetration of portable devices and their short communication range (e.g., 10 meters for Bluetooth Class 3)

[3] Note that PeopleNet with its original setting (where people are disseminating queries) will not suffer from this problem since encounters among people are not frequent, and portable devices usually have small buffers compared to the number of incoming requests.

carries packets and forwards them to a newly found vehicle that is moving towards the destination. This works well only for *delay tolerant* applications [16]. [6] uses the knowledge of the relative velocities and directions of one's neighbors to make a forwarding decision. MDDV [12] utilizes vehicle traffic history data and route packets based on a predetermined trajectory, using a digital map. VADD [15] monitors vehicle position/motion information within a bounded area to choose the packet forwarding direction (i.e., road selection at an intersection) that minimizes the delivery delay

A prerequisite of geographic routing is a location service that tells where the destination is. Devising efficient, scalable, and robust location services has been an active area of research in recent years [14]. Unfortunately, the frequent changes experienced in a vehicle network topology make accurate location service costly. An elegant way of reducing this cost is by exploiting spatial-temporal correlation that exists in most realistic mobility patterns; i.e., the distance between two nodes is more or less correlated with the time elapsed since they last encountered each other. This observation brought forth Last Encounter Routing (LER) [4]. In FleaNet, since each vehicle can piggy-back the current position into its query advertisement, LER can be supported at no extra cost. LER, however, does not address intermittent connectivity. In this paper, we enhance LER by providing the carry-and-forward functionality. As we will see later, enhanced LER plays a key role in FleaNet when the query source needs to notify its decision to the owner of the chosen match.

III. FleaNet System

In this section, we describe the FleaNet system. We first give an overview of the FleaNet architecture and underlying vehicular networks. We then present the FleaNet query dissemination/matching notification protocol.

A. FleaNet architecture

The goal of FleaNet is to exchange/resolve queries and conclude transactions (e.g., sell/buy a baseball ticket near the stadium before the play time) in urban vehicular networks where vehicles communicate through a wireless interface such as Dedicated Short Range Communication (DSRC) [2]. DSRC devices operate in the 5.9 GHz band and can achieve up to 27 Mbps within a range of 1000m. Vehicles as well as static roadside *Adstations* (see Figure 1) generate and propagate queries. Adstations can be stores advertising their products. For example, a pizzeria could advertise its special pizza offer to vehicles passing by and a driver who received the advertisement could place an order.

Unlike typical personal mobile devices such as PDAs, FleaNet nodes (e.g., in-car computers, desktop machines in a building) are not restricted by the energy constraints and thus, are equipped with fairly high processing power and large storage. On top of this storage, we assume that a light-weight database such as Berkeley DB [1] is running to store and resolve queries from other vehicles. FleaNet does not consider the Internet connection through the hotspots available

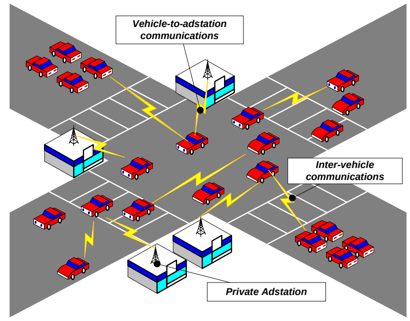


Fig. 1. Example of FleaNet

on the urban streets, but rather focuses on query exchange and resolution among vehicles. Moreover, we simply assume that, with the same goal of query exchanging and resolution, users are willing to cooperate.

B. FleaNet protocol

The FleaNet protocol is composed of query dissemination and match/transaction notification. We discuss message formats first and then describe the protocol in detail. The efficacy of the proposed protocol is analyzed in Section IV.

1) Message formats: We define message formats used in FleaNet and explain each message type in detail. Every message has the common message header which includes a message type field followed by user ID (UID) and encoded positions (see Figure 2).[4] FleaNet defines three message types: query, match, and transaction. First, a query message is used for representing the information about the goods that a person possesses or seeks. Second, a match message is used to inform users that a node finds queries with a matching interest. Finally, a transaction message is used whenever a user wants to make a transaction request or respond to such a request. Let us look at each message type in more detail.

A query includes sequence number, query type, notification flag (N), maximum number of matches (N_{MAX}), reference location/radius (R-Loc/Radius), expiration time, multi-level description ($L_1 \cdots L_N$), and additional description (see Figure 2). Sequence number along with the user ID is used to uniquely identify a query. There could be potentially many types of queries, but for the sake of simplicity, in this paper we assume that there are two query types: buy and sell, both for the same item. A notification flag is set if a user wants to receive the query resolution results. By setting N_{MAX}, a user can explicitly set the maximum number of matches that a user wants to receive. R-Loc contains the user's reference location, and the dissemination radius. A user can define an arbitrary location (e.g., one's real home address) as home location and accordingly set the dissemination radius. The expiration time of a query can be set through Exp. The multi-level description is given based on a hierarchical order. Note that the N_{MAX}

[4] We represent a position four byte floating point quantities, for x and y coordinates, which is used for last encounter routing as described in the previous section.

or the expiration time field permits the nodes to automatically dispose the query after either receiving N_{MAX} matches or passing the expiration time. With regard to the multi-level description field, we simply assume that FleaNet users are given the detailed version of an eBay-like hierarchy. A user could leave additional comments in the user description (UD) field. Here, we assume that users' descriptions of an item are uniform; e.g., users who are interested in "iPod Nano" will always describe it using the same sentence. Note that since query size is relatively small, a node typically packs a number of queries into a single packet. Figure 2 shows that N queries are packed into a single packet.

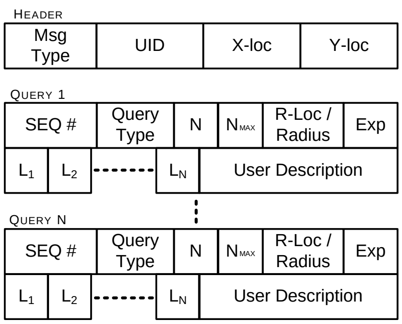


Fig. 2. A set of N queries in a single packet

A match occurs when two different types of queries (e.g., buy and sell queries) have the same hierarchical structure as well as user description. For example, given that we have query a and b, a match happens when $a.L1 = b.L1$, $a.L2 = b.L2$, $\cdots$, $a.LN = b.LN$, and $a.UD = b.UD$. When a node finds a match, it replies with a match message which includes the user ID and the sequence number of the matched query. Note that using complex similarity measures such as [5] is beyond the scope of this paper. After this, a person can make his final decision by sending a transaction request which includes the UID and sequence number of the matching query. Upon receiving the request, the owner of the matching query responds with a transaction response through which he will notify his decision on whether to accept or reject the offer.

Example 1: One day Joe Bruin wants to sell some of his items, but he is too busy with his work to do a garage sale. In this situation, FleaNet helps him to sell the items while he is behind the wheel (i.e., mobile garage sale!). He inputs details of the items using FleaNet software to create queries of items; for example, L1="Consumer Electronics," L2="MP3 Players," L3="Apple iPod," UD="iPod Mini, 4G." Since he is commuting between downtown LA and west LA, he wants to find buyers near that area. Using a digital map provided by FleaNet software, he can easily set the area of interest to which his queries will be disseminated. For some items, he wants to see multiple matches, say 5, to make the best deal by simply setting $N_{MAX} = 5$. He also wants to sell the items while he is commuting, which takes about half an hour, and thus, he sets the expiration time accordingly. As a result, this query will be advertised and is spreading near his commuting path through vehicular networks using the query dissemination protocol described in Section III-B.2. Some time later, the query will be responded with a match message (i.e., a sell query of a ticket). Joe Bruin will then send a transaction request message to sell his item, and in the end, he will receive a transaction response from the originator of the matched query.

2) Query dissemination: A node (mobile/static) periodically broadcasts its query using QUERYBROADCAST to its one-hop neighbors. We use random jitter to avoid packet collisions due to synchronized broadcasting among neighbors. Each node listens to its neighbors' query broadcasts and stores the received queries into its local database. Owing to the nodes' mobility, queries are opportunistically disseminated into the entire network. Figure 3 depicts the case of two mobile nodes (i.e., C_1 and C_2) that encounter with other nodes over time. A black triangle with timestamp in the figure represents an encounter between C_1 and other nodes which are within the communication range of C_1 at that time. We denote the query generated by node C_k as Q_{C_k}. Since C_1 and C_2 periodically advertises a query Q_{C_1} and Q_{C_2} respectively, when they meet each other at time $T - t_4$ they can receive and store each other's query. In other words, C_1 carries Q_{C_2} and C_2 carries Q_{C_1} in their local database after time $T - t_4$.

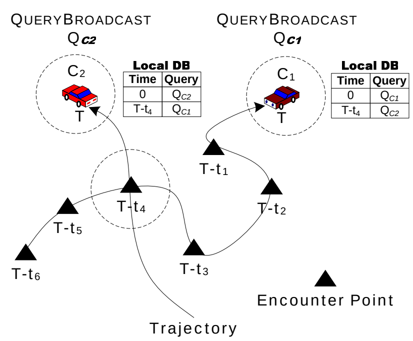
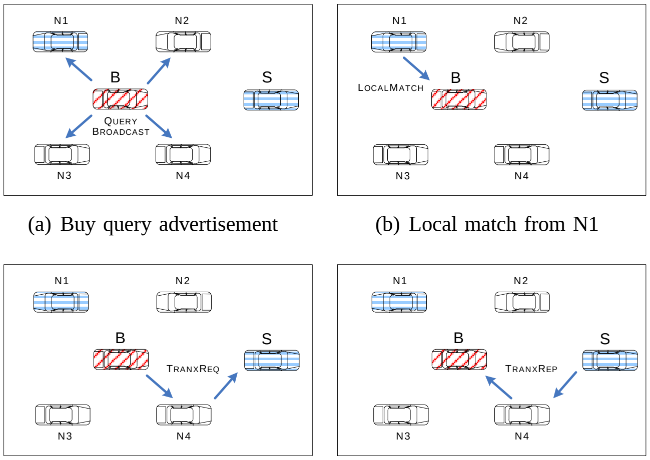


Fig. 3. Query dissemination

3) Match and transaction notification: Every incoming query is resolved from the local database. If a node C_{RES} finds a set of matched queries $Q_{\acute{C}_1}, Q_{\acute{C}_2}, \cdots, Q_{\acute{C}_k}$ to an incoming query $Q_{C_{IN}}$, this set of matched queries will then be sent using LOCALMATCH to the query originator, C_{IN}. The resolver notifies the results only to C_{IN}, and it does not notify the results to the originators of the matched queries, i.e., $\acute{C}_1, \acute{C}_2, \cdots, \acute{C}_k$. If the number of matched queries, k is larger than N_{MAX} (i.e., the maximum number of matches that C_{IN} wants to receive), then the resolver will randomly pick N_{MAX} number of matches and send them to C_{IN}. Otherwise, k matched queries are returned to C_{IN}. After this, the query originator, C_{IN} updates N_{MAX} field of the query by subtracting k. If N_{max} goes below zero, then the query will be discarded. Finally, C_{IN} will choose one of the matched queries and will notify his decision to the originator of the chosen matched query, say $\acute{C}_\ell$, by sending a transaction request message, i.e., TRANXREQ. If $\acute{C}_\ell$ accepts the transaction, then he will respond with a transaction reply



(a) Buy query advertisement (b) Local match from N1

(c) Transaction request (d) Transaction response

Fig. 4. Match and transaction notification

message, i.e., TRANXREP.

Figure 4 shows an example of match notification. Let us assume that node B and node S is advertising sell and buy query respectively, and $N1$ is carrying the sell query of node S since node $N1$ has already met S. In Figure 4(a), node B advertises its buy query to its neighbors. Node $N1$ then finds a local match and sends LOCALMATCH to node B as shown in Figure 4(b). As a result, node B makes its final decision by sending TRANXREQ to node S and thus, S will respond with TRANXREQ to node B as shown in Figure 4(c) and Figure 4(d).

IV. FEASIBILITY ANALYSIS

In this section, we first develop an analytical model for the notification latency. Namely, for a given query, how long does it take until the originator is notified of a match? We then derive a second model to demonstrate FleaNet's scalability

A. Notification latency

Let us assume that in an area $L \times L$ meter square, there are N nodes each of which communicates with other nodes within a radio range of R meters. We also assume for simplicity that there exists a single target query of interest (e.g., 1 buyer and 1 seller). Given that nodes are moving based on random waypoint, random direction, or Manhattan mobility, [3] shows that the inter-meeting time between two mobile nodes follows an exponential distribution. This is also true when nodes are moving on an irregular urban grid.

Let us first characterize the match latency defined as the time for a random seller to meet one of the nodes with a matching query, i.e., either the buyer itself or any node that encountered the buyer. The same definition holds for a buyer seeking sellers. According to [3], the average latency D can be expressed as follows:

$$\mathbb{E}[D] = \frac{1}{\lambda}\Big(\sqrt{\frac{\pi}{2N}} + \mathcal{O}(\frac{1}{N})\Big) \quad (1)$$

Here, λ depends on the mobility pattern and is defined as $\lambda \approx \frac{2\alpha r \mathbb{E}[\mathbb{V}^*]}{L^2}$ where $\mathbb{E}[\mathbb{V}^*]$ is the average relative speed between two mobile nodes. From Equation 1, we see that the latency is inversely proportional to the number of nodes and to the rate λ (and thus average relative speed). Fast mobility in urban vehicular networks reduces the average latency. However, as we will see in the simulation section, the restricted mobility patterns in the urban environment severely decrease the odds of meeting (compared to random mobility), thus offsetting the benefits of fast mobility.

It is interesting to note that a match can happen at both sides. If a match happens at the buyer side, the buyer sends a notification to the remote seller. One may wonder if by sending a notification to the remote buyer one could potentially reduce the average notification latency. By the following argument we show that there is no difference in terms of the average notification latency. Let D_S and D_B denote the random variables of the seller's and buyer's match latencies respectively. The notification latency at the seller is the minimum of the two random variables, $D_{notify} = \min\{D_S, D_B\}$. Then, the average is given as $\mathbb{E}[D_{notify}] = \min\{\mathbb{E}[D_S], \mathbb{E}[D_B]\}$. Since all nodes are independent each other, we see that D_S and D_B are IID random variables. From this, the average notification latency is expressed as $\mathbb{E}[D_{notify}] = \mathbb{E}[D_S] = \mathbb{E}[D_B]$. Therefore, a match at the other side does not give us any benefit in terms of average latency *if both processes start at the same time*.

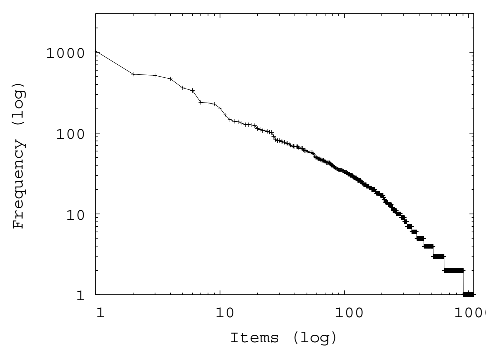


Fig. 5. Posted item popularity distribution (log-log plot)

In reality, it is likely that there will be many people with the same interest. Figure 5 shows the popularity distribution of 16,862 postings (make+model) in the vehicle ad section of Craigslist[5] during March 1-7, 2006. From the figure, we see that it approximately follows the power-law distribution. The top 100 items take up 60% of the total advertisements. Given such a distribution, let us analyze the impact of popularity. Assume that there are K users with the same interest (e.g., selling "iPod nano"), which is equivalent to K query advertisers. To characterize the impact, we need to derive the distribution that a random node meets one of K nodes. This is the minimum of K exponential random variables that is simply expressed as an exponential random variable with parameter $K\lambda$. In other

[5] http://www.craigslist.org/

words, the rate of query dissemination has increased K times. Therefore, the average latency with K advertisers is simply given as $\mathbb{E}[D]/K$ (see [7] for detail derivations). We conclude that popularity of a query can greatly improve the latency.

B. Scalability

FleaNet must be scalable such that it can operate in a large scale VANET. The scalability analysis of our query dissemination protocol can be addressed by characterizing the induced traffic load (i.e., channel bandwidth consumption). Every node broadcasts a query packet to its neighbors once every T_a seconds where T_a is the inter-broadcast duration. So, the number of packets received by a node is bounded by the number of nodes met during T_a seconds. Recall that in our epidemic model a node upon hearing a query does not relay it again. Further propagation occurs because of motion. The number depends on node density but not on overall number of nodes. In contrast, any “flooding”-based dissemination protocol is not scalable because a node could potentially receive a number of packets proportional to network size. This is because in flooding each node repeats the query to its neighbors. In fact, in a static network, this is the only way propagation can take place.

To give a rough idea of the traffic generated by query dissemination, let us simply use $T_a = \frac{2R}{V^*}$ (i.e., the time for a mobile node travels the communication diameter) where R is the transmission range and V^* denotes the relative speed of two mobile nodes.[6] We intentionally use the maximum relative speed to consider the worst case scenario. Assuming that nodes are moving at the speed of $V = 5m/s$, the maximum relative speed is given as $V^* = 10m/s$. Let us also assume that we have the transmission range $R = 250m$, the fixed packet size $P = 1500B$, and the link speed $S = 11Mbps$. Thus, the advertisement interval is given as $T_a = \frac{250m}{10m/s} = 25s$, and the transmission time for one packet is about $1ms$. While traveling for T_a, a regular node will be exposed to advertisements from an area[7] of $\mathcal{A} = \pi R^2 + 4R^2$. In the worst case, all nodes within this area are distinct and potentially send their generated packets to the considered node (potential senders $n = \mathcal{A}\rho$). The worst case link bandwidth consumption is estimated as nT_x/T_a. For instance, given a relatively high populated area with $N = 1,000$, the number of potential senders is $n \simeq 90$, and the FleaNet protocol consumes in the worst case only 0.0035 of the link bandwidth.

[6]For a given speed, we want to make sure that the interval is neither too short nor too long compared to the average connection duration among nodes. If it is too short, then we are unnecessary sending out more packets to the same set of nodes, thus increasing link bandwidth utilization. On the other hand, if it is too long, a node misses chances to send the packets to encountered nodes, thus slowing down dissemination. To this reason, we simply set the value $T_a = \frac{2R}{V^*}$ as the time for a mobile node travels the communication diameter (2R).

[7]Note that if every node is stationary, the exposed area is simply πR^2. In our case, since nodes are moving, a mobile node actually encounters more nodes than a stationary node.

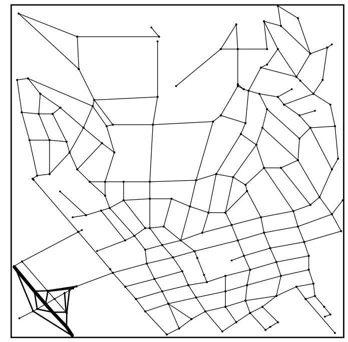


Fig. 6. Westwood area in the vicinity of UCLA

V. Evaluation

In this section, we evaluate FleaNet performance through extensive simulations using Ns-2 [10].

A. Simulation setup

Simulation results are obtained by considering vehicle nodes with IEEE 802.11 connectivity with 2Mbps bandwidth, 250m radio range, and two-ray ground reflection model for radio propagation. Nodes move according to the Real-Track (RT) mobility model [17]. RT permits to model vehicle mobility in an urban environment more realistically than other simpler and widely used mobility models, such as Random Way Point (RWP) by restricting the areas where nodes can appear (e.g., roads). Also, in the RT model vehicles tend to aggregate and move in groups because of traffic signals and because directions change only at road intersections. Our simulations consider a vehicular network with a number of nodes between 100 and 300. Vehicles move in the $2,400m \times 2,400m$ $Westwood$ area in the vicinity of the UCLA campus. The map was obtained from the US Census Bureau data for street-level maps and is shown in Figure 6. Vehicles travel with an average speed between 5 m/s and 25 m/s. For example, if we set RT with v_{min}=0 and v_{max}=10, we have an average speed of 5 m/s.

To evaluate the performance of the system, we use the average latency of notification as the metric. For a given query, the latency measures the time for a node to receive a notification. This happens when either it meets a node that matches the query (i.e., LocalMatch) or a it receives a notification of a remote match, i.e., TranxReq. The latency is dependent on many parameters, specifically on density/speed, query popularity, and mobility. In this section, we focus on investigating the impacts of such parameters.

B. Impact of density and speed

Our analysis in Section 4.1 shows that the average latency is a function of node density and speed. In this section, we study the effect of restricted mobility on the average latency.

Figure 7 shows the latency as a function of density and speed in case of one seller and one buyer—a seller and a buyer are randomly picked from the nodes in the network in each simulation. X-axis and Y-axis represent the average vehicle

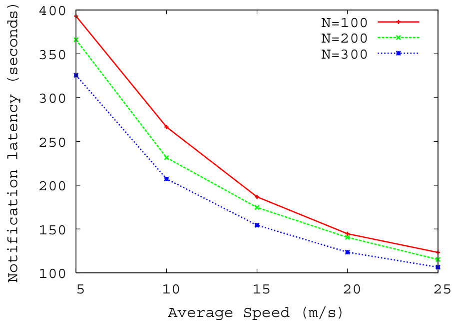


Fig. 7. Average latency as a function of speed

speed varying from 5 m/s to 25 m/s and the matching latency respectively. From the figure, we see that both the density and the speed of vehicles are important factors in determining the matching latency. As the average speed or the number of nodes increases, the average matching latency decreases. This is not surprising since, intuitively, as the average speed or the number of nodes increases, a node has higher chance of meeting other nodes, which translates into more rapid dissemination of the query and also a higher chance to find a match.

The figure also shows that the relative benefit of increasing the number of nodes reduces as nodes' speed grows. For example, in the case of 5 m/s, if we increase the number of nodes from 100 to 300, we obtain a 20% latency improvement. On the other hand, the case of 25 m/s only shows a 13% improvement. This trend is consistent with our analysis in Section IV-A, indicating that the average speed is the dominating factor in the match latency. We have shown that the latency is inversely proportional to the average speed and to the square root of the number of nodes (Equation 1). Note also that if nodes' speed is fixed, even by increasing the density, it is not possible to decrease the average latency below a certain threshold. This threshold corresponds to the average time for a random buyer to travel the distance to a random seller and is dependent exclusively on the speed.

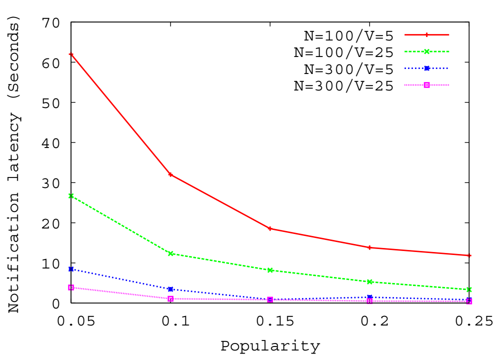


Fig. 8. Impact of delay with respect to the popularity

C. *Impact of query popularity*

The notification latency of a query is heavily dependent on the popularity of the query. We can easily see that if many people are interested in a specific item, a notification on the item will quickly happen. Figure 8 confirms our intuition on the impact of popularity on the latency: as the popularity increases, the latency decreases. The figure plots the latency as a function of popularity in a single-buyer, k-seller case. In the figure, we increase ratio of sellers in the network from 5% to 25% with a gap of 5% on X-axis and vary the average speed (i.e., 5/25 m/s) and the number of nodes (i.e., 100/300 nodes). Given a single buyer, k% sellers are randomly selected and we measure the latency for a notification. We limit ourselves to the single buyer case to clearly see the impact of query popularity on the latency. The figure also show that when the number of nodes is small (say, N=100) and mobility is slow (say, V=5m/s), we see that popularity greatly improves the overall latency.

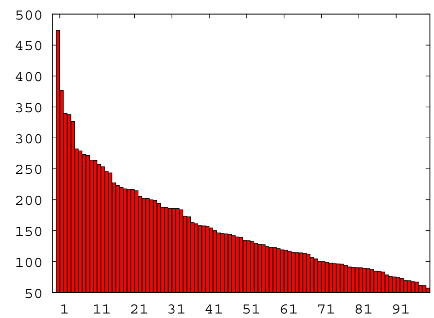


Fig. 9. Impact of location on matching latency

D. *Impact of location*

Nodes can be static, e.g., Adstation, in FleaNet. In this section, we show how a stationary node affects the average notification latency, i.e., the impact of its location. Intuitively, since the average relative speed of two nodes is higher if both move, a mobile node has higher chance of meeting more nodes than stationary one, which results in faster query dissemination. Restricted mobility in the real-track mobility model makes the situation of a stationary node worse, because nodes tend to stay longer in the area where roads are densely clustered together. This can be better understood by modeling the travel of a vehicle through the urban grid as a Markov chain, where each state in the Markov chain represents a vehicle occupying a given road segment. The vehicle moves from segment to segment. Assuming that a vehicle at the end of a segment chooses its direction randomly among k forks, i.e., the current segment is connected to k different road segments, the transition probability towards each is simply $1/k$. For ease of illustration, we assume that each segment has the same number of forks; and, the same "residence" time (i.e., the vehicle spends the same average time in each segment). Then, the probability that a vehicle stays in each segment is simply $1/N$ where N is the number of segments. And, the probability that a vehicle stays in an area with m segments is simply m/N. From this we conclude that the thicker the concentration of segments in the area (e.g., historic downtown), the higher the probability that a vehicle stays in that area. In reality, cars will traverse short segments faster than long segments; still, the simulation experiments show that on average there is higher persistence in dense areas.[8]

In order to clearly understand how the position of a stationary node affects the latency, we used a scenario with 100 nodes moving at the average speed of 25 m/s. We intentionally fixed the location of a random node to its initial position which is uniformly distributed in the simulated area. We set the fixed node as a buyer and rotate the buyer role on all the 100 nodes. As a result, for a given scenario file (N=100, V=25m/s), we generated 100 scenario files by fixing a node one by one to its initial position. Thus, for a given buyer (i.e., a scenario file) we performed 99 trials and get the average latency, by selecting each one of the other 99 nodes as a seller.

In Figure 9, we plot the latency distribution for the entire node population. Here, the ith index of X-axis represents a node with ith largest latency. The figure shows that the latency heavily depends on the location: for example, the best location (Rank 100) has an average latency of 57.139 seconds whereas the worst location (Rank 1) has 473.856 seconds (8.3 times larger!). By examining the location of the static node in each experiment we found that, as expected, the location of node ranked 100, i.e. with the smallest latency, is in the middle of the area where roads are densely clustered together whereas that of node ranked 1, i.e. with the largest latency, is in the northwest area where roads are sparse. Note that the average latency calculated over all stationary nodes is 156.9 seconds, whereas that of mobile nodes is 123.25 seconds; as expected, mobile nodes perform better than stationary ones.

VI. Conclusion

In this paper we have proposed a virtual market place concept where a mix of mobile and stationary users can carry out buy/sell transactions (or any other matching of common interests) using a vehicular network. To realize this concept, we have shown the FleaNet architecture which defines the details of FleaNet components such as FleaNet nodes (e.g., Adstations) and query formats. Query dissemination and resolution are carried out through a FleaNet protocol suite which is scalable up to thousands of nodes and non-intrusive to other existing services. We have evaluated the proposed protocols via analysis and simulations. Our results show that in most cases, a random query can be resolved within a tolerable amount of time, and location of a stationary advertiser must match the urban grid layout for low latency query resolution.

Acknowledgment

We would like to thank Biao Zhou for allowing us to use mobility scenarios and Eugenio Magistretti for sparing his valuable time to review the paper.

[8] For ease of analysis, we simply show the number average using a Markov chain. Readers can find a more detailed model with a semi Markov chain in the extend version of this paper [7].

References

[1] Berkeley DB. `http://www.sleepycat.com/`.

[2] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Sept. 2003.

[3] R. Groenevelt, P. Nain, and G. Koole. Message delay in Mobile Ad Hoc Networks. In *Performance'05*, Juan-les-Pins, France, Oct. 2005.

[4] M. Grossglauser and M. Vetterli. Locating Nodes with EASE: Mobility Diffusion of Last Encounters in Ad Hoc Networks. In *IEEE INFOCOM*, San Francisco, CA, USA, Mar.-Apr. 2003.

[5] T. Hofmann. Probabilistic Latent Semantic Analysis. In *UAI'99*, 1999.

[6] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang. Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks. In *IEEE VTC'05*, Dollas, TX, USA, Sept. 2005.

[7] U. Lee, J.-S. Park, E. Magistretti, E. Amir, and M. Gerla. FleaNet: A Virtual Market Place on Vehicular Networks. Technical report, UCLA, May 2006.

[8] M. Motani, V. Srinivasan, and P. S. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *MobiCom'05*, Aug. 2005.

[9] A. Nandan, S. Das, S. Tewari, M. Gerla, and L. Klienrock. AdTorrent: Delivering Location Cognizant Advertisements to Car Networks. In *WONS'06*, Les Menuires, France, Jan. 2006.

[10] ns-2 (The Network Simulator). `http://www.isi.edu/nsnam/ns/`.

[11] H. Rheingold. *Smart Mobs: The Next Social Revolution*. Basic Books, 2003.

[12] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. MDDV: a Mobility-entric Data Dissemination Algorithm for Vehicular Networks. In *VANET'04*, Philadelphia, PA, USA, Oct. 2004.

[13] B. Xu, A. Ouksel, and O. Wolfson. Opportunistic Resource Exchange in Inter-vehicle Ad-hoc Networks. In *IEEE/ACM MDM*, Berkeley, CA, USA, Jan. 2004.

[14] Y. Yu, G.-H. Lu, and Z.-L. Zhang. Enhancing Location Service Scalability with HIGH-GRADE. In *IEEE MASS'04*, Fort Lauderdale, Florida, USA, October 2004.

[15] J. Zhao and G. Cao. VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks. In *IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[16] W. Zhao, M. Ammar, and E. Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *ACM MOBIHOC'04*, Tokyo, Japan, May 2004.

[17] B. Zhou, K. Xu, and M. Gerla. Group and Swarm Mobility Models for Ad Hoc Network Scenarios Using Virtual Tracks. In *IEEE MILCOM*, Monterey, CA, USA, Oct.-Nov. 2004.